4

Contractor Report ARFSD-CR-90016

An Evaluation of Shape Methods for Helicopter Classification and Orientation Determination

Timothy A. Grogan
University of Cincinnati
Cincinnati, OH 45221

Frank P. Kuhl
ARDEC Project Engineer

September 1990

US ARMY
ARMAMENT MUNITIONS
& CHEMICAL COMMAND
ARMAMENT RDE CENTER

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT AND ENGINEERING CENTER

Fire Support Armament Center
Picatinny Arsenal, New Jersey

Approved for pullic release;
Distribution is unlimited.

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

The citation in this report of the names of commercial firms or commercially available products or services does not constitute official endorsement by or approval of the U.S. Government.

Destroy this report when no longer needed by any method that will prevent disclosure of contents or reconstruction of the document. Do not return to the originator.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION DOWNGRADING SCHEDULE

3. DISTRIBUTION AVAILABILITY OF REPORT
Approved for public release;
Distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER

5. MONITORING ORGANIZATION REPORT NUMBER
Contractor Report ARFSD-CR-90016

6a. NAME OF PERFORMING ORGANIZATION
T. Grogan

6b. OFFICE SYMBOL

7a. NAME OF MONITORING ORGANIZATION
ARDEC, FSAC

6c. ADDRESS (CITY, STATE, AND ZIP CODE)
University of Cincinnati
Cincinnati OH 45221

7b. ADDRESS (CITY, STATE, AND ZIP CODE)
Fire Control Division
Picatinny Arsenal, N. J. 07806-5000

8a. NAME OF FUNDING SPONSORING ORGANIZATION
ARDEC, IMD STINFO BR

8b. OFFICE SYMBOL
SMCAR-IMI-I

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DAAG29-81-D-0100

8c. ADDRESS (CITY, STATE, AND ZIP CODE)
Picatinny Arsenal, N. J. 07806-5000

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (INCLUDE SECURITY CLASSIFICATION)
An Evaluation of Shape Methods for Helicopter Classification and Orientation Determination

12. PERSONAL AUTHOR(S)
Timothy A. Grogan, University of Cincinnati and Frank P. Kuhl, ARDEC Project Engineer

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 7/7/87 TO 3/31/88

14. DATE OF REPORT (YEAR, MONTH, DAY)
September,1990

15. PAGE COUNT
93

16. SUPPLEMENTARY NOTATION
Task was performed under a Scientific Services Agreement issued by Battelle, Research Triangle Park Office
200 Park Drive, P.O. Box 12297, Research Triangle Park, N.C. 27709

17. COSATI CODES

FIELD	GROUP	SUB GROUP

18. SUBJECT TERMS (CONTINUE ON REVERSE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)
Shape recognition, Fourier descriptors, Walsh points, Rotation transformation, Orientation determination, Helicopter fuselage orientation.

19. ABSTRACT (CONTINUE ON REVERSE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)

The performance of two competing methods are evaluated for classification and orientation determination accuracy. Helicopter fuselage silhouettes are classified as to their type and orientation. The methods are tested for sensitivity to imaging noise and the reference library sampling density.

Under all conditions of imaging noise and library sampling density, the Fourier descriptor method out performs the Walsh points method for helicopter type classification. Helicopter type classification performance for the Fourier method ranges from 99% to 84% even under the most severe conditions examined. In almost all cases, the Fourier method exhibits median angle errors one half those of the Walsh points method. Median angle errors of one half the library sampling interval have been attained using the Fourier descriptor method.

A scheme for helicopter rotor segmentation and orientation determination including the necessary additional library enhancements is also proposed.

20. DISTRIBUTION AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED UNLIMITED ☒ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
I. HAZNEDARI

22b. TELEPHONE (INCLUDE AREA CODE)
AV 880-3316

22c. OFFICE SYMBOL
SMCAR-IMI-I

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

Contents

List of Figures

List of Tables

Introduction

The purpose of this effort was to investigate the sensitivity of various shape recognition algorithms to the effects of shape library sampling density and imaging noise as applied to the task of helicopter classification and orientation determination.

Section 1 introduces the first of two shape methods, Fourier descriptors, whose performance is evaluated. This section discusses the basic method as well as the algorithms for the efficient computation and normalization of the shape method feature vector.

Section 2 introduces the second shape method, Walsh points. The definition of this shape method is stated. Then the method for computation and normalization are discussed.

Section 3 discusses the procedure used to generate the multiple views of the reference shapes for inclusion into the reference shape feature vector libraries.

Section 4 describes the two sets of experiments to determine the performance of the two algorithms for helicopter fuselage type classification and orientation determination. The goal of the experiments is to determine the sensitivity to library sampling density and imaging noise.

Section 5 interprets the results of the experiments described in Section 4.

Section 6 consists of a proposed scheme for the library procedures, segmentation, and detection of the helicopter tip path for rotor blade orientation determination.

Section 7 provides the development of the transformations for determining the "unknown" helicopter orientation given the orientation of the imaging system for a non-earth centered imaging platform.

The results of the entire study are summarized in the Conclusion.

The first appendices provides the raw classification and angle error information including the classification confusion matrices for all the experimental conditions examined in the study.

The last appendix is a listing of the equations required to compute the elements of the observed object orientation relative to the camera reference frame.

1 Fourier Descriptors

1.1 Introduction

The first method used in recognizing the helicopter fuseluge silhouettes is based on the well known method of Fourier series analysis [3], [4], [6], [10], [15], [16], [17]. Having been provided with the silhouette of an object, the contour or boundary completely specifies the two-dimensional shape. The contour can be parameterized as a function of time by tracing around the boundary in counterclockwise direction. As the tracing continues the function begins to repeat. Since this function is periodic, it can be expanded into a Fourier series. Each basis function of the complex exponential Fourier series is non-zero almost everywhere over each period. This is a global method of shape analysis. Therefore, when segmentation errors become increasing large, the recognition accuracy degrades.

The boundary function, γ, maps the real number line into the complex plane. The projection on the real axis is the x component and the orthogonal projection on the imaginary axis is the y component.

$$\gamma(t) = x(t) + iy(t) \subset \mathbb{C}.$$

The velocity of the tracing is $v(t) = \frac{d\gamma(t)}{dt}$. So, the speed of tracing is equal to the magnitude of the velocity, i.e.

$$\mid v(t) \mid^2 = \frac{d\gamma(t)}{dt}\frac{d\gamma^*(t)}{dt}.$$

If we assume that the tracing takes place at a constant speed (*uniform tracing*), then

$$v = \mid \frac{d\gamma(t)}{dt} \mid = \left(\left(\frac{dx(t)}{dt} \right)^2 + \left(\frac{dy(t)}{dt} \right)^2 \right)^{1/2} = constant.$$

Since the speed is constant and the period of the trace is T, then $T = v$L, where L is the total arc length once around the contour. Assuming $\gamma(t)$ is a continuous, bounded, periodic function with finite arc length (i.e. rectifiable), $\gamma(t)$ can be expanded in the Fourier series

$$\gamma(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i\frac{2\pi n}{T}t},$$

where

$$c_n = \frac{1}{T}\int_0^T \gamma(t) e^{-i\frac{2\pi n}{T}t} dt.$$

1.2 Calculation

Since the images are discrete, connecting the contours of the pixels around the boundary produces a polygon. The Fourier coefficients can be computed directly from the increments in time (arc length) and position as the polygon is traced. This is called the direct Fourier transform (DFT). The following derivation follows along the lines of [6], [10].

Let the increment in position be the complex number $\Delta\gamma_i$, and the increment in time will be $\Delta t_i = \mid \Delta\gamma_i \mid = \sqrt{\Delta x_i{}^2 + \Delta y_i{}^2}$. Since γ is piecewise linear, its derivative, $\dot{\gamma}$, is piecewise constant. So, let

$$\dot{\gamma}(t) = \sum_{n=-\infty}^{+\infty} \beta_n e^{i\frac{2\pi n}{T}t}$$

be the Fourier series for $\dot{\gamma}$. But the series for γ can be differentiated term by term so that

$$\dot{\gamma} = \sum_{n=-\infty}^{+\infty} i\frac{2\pi n}{T} c_n \, e^{i\frac{2\pi n}{T}t}$$

So,

$$\beta_n = i\frac{2\pi}{T} c_n \Longrightarrow c_n = -i\frac{T}{2\pi n}\beta_n, \; n \neq 0.$$

Now,

$$\beta_n = \frac{1}{T}\sum_{p=1}^{K} \frac{\Delta\gamma_p}{\Delta t_p}(e^{-i\frac{2\pi}{T}t_p} - e^{-i\frac{2\pi}{T}t_{p-1}})\;(i\frac{T}{2\pi n})$$

We can then write

$$c_n = \frac{T}{4\pi^2 n^2}\sum_{p=1}^{K}\frac{\Delta\gamma_p}{\Delta t_p}\left(e^{-i\frac{2\pi}{T}t_p} - e^{-i\frac{2\pi}{T}t_{p-1}}\right), \; n \neq 0, \qquad (1)$$

where

$$t_p = \sum_{i=1}^{p}\Delta t_i, \;\; t_0 = 0,$$

a_i :	0	1	2	3	4	5	6	7
$\Delta\gamma$:	1	1+i	i	-1+i	-1	-1-i	-i	1-i
Δt :	1	$\sqrt{2}$	1	$\sqrt{2}$	1	$\sqrt{2}$	1	$\sqrt{2}$

Table 1: Contour line increments for the 8-direction chain code.

K is the number of sides on the polygon, and $T = t_K =$ period. For the c_0 term,

$$c_0 = \frac{1}{T}\sum_{p=1}^{K}(1/2\Delta\gamma_p + \gamma_{p-1})\Delta t_p. \tag{2}$$

For the discrete images, the contours are represented by the eight-direction chain code. So, the $\Delta\gamma_i$ and Δt_i can be determined using a lookup table indexed by the chain code.

The computational complexity is of order NK, where N is the number of coefficients calculated and K is the number of links in the chain coded contour.

1.3 Normalization

It is usually desirable to compare shapes independent of size, orientation, and starting point. To compare two shape boundaries using their Fourier descriptors, it is necessary to scale, rotate, and shift their shapes in order to allow the "best" fit possible. This operation normalizes the Fourier descriptors for the unknown shape to an optimum orientation. It has been shown [15] that the optimum scale, rotation angle, and relative starting point shift can be obtained to minimize the mean-squared error as the criterion.

To reduce the computational burden to normalize the coefficients, many others have suggested that one of several suboptimum methods be used. Instead of normalizing the coefficients of the unknown differently for each template, the coefficients are normalized to a "standard" orientation independent of the template.

First the object is translated to the origin by setting $c_0 = 0$. Because the fundamental or c_1 has always been observed to be the largest ($n \neq 0$), all the coefficients are scaled by dividing by $\mid c_1 \mid$. Next, the shape is rotated and the starting point shifted. Most of the variations among the suboptimal methods described in the literature occur in how this rotation and starting point shift is obtained [12], [13], [16], [17].

The following algorithm is used in this study to normalize the shape coefficients:

1. Set $c'_0 = 0$.

2. Scale so that $| c'_1 | = 1$.

3. Find the coefficient c_k that is next largest ($| k - 1 | \leq 5$, $k \neq 0, 1$.) If $| k - 1 | > 5$, then let $k = 2$.

4. Rotate and shift the starting point so that $\angle c'_1 = 0$ and $\angle c'_k = 0$, i.e.

$$c'_n = \frac{c_n}{| c_1 |} e^{i(nt_0 + \alpha)}, \text{ where } t_0 = \frac{\angle c_1 - \angle c_k}{k - 1}, \quad \alpha = -\frac{(k\angle c_1 - \angle c_k)}{k - 1}.$$

The object $\gamma(t) = c_1 e^{i\frac{2\pi}{T}t} + c_k e^{i\frac{2\pi k}{T}t}$ has $| k - 1 |$-fold symmetry. So, there are $| k - 1 |$ rotations and relative starting point shifts, multiples of $\frac{2\pi}{k-1}$ that will satisfy the zero phase condition. So, rotate and shift the starting point so that

$$\tilde{c_n}(m) = c'_n e^{in\frac{2\pi m}{k-1} - i\frac{2\pi m}{k-1}}, \quad m = 0, 1, \ldots, (| k - 1 | -2).$$

to maximize the following function:

$$\sum_n \Re\{\tilde{c_n}(m)\} \mid \Re\{\tilde{c_n}(m)\} \mid .$$

This criterion effectively chooses the normalization that orients the contour so that the axis of one of the main lobes of the $| k - 1 |$-fold fundamental shape $c_1 e^{it} + c_k e^{ikt}$ is along the positive x-axis and the starting point on the contour corresponding to that lobe where the contour is farthest from the origin. In order to reduce the number of rotations to perform and calculate the criterion only, $| k - 1 | \leq 5$ is allowed. If $| k - 1 | > 5$, then $k = -1$.

The correct recognition of a shape is very sensitive to this rotation and starting point normalization. In order to improve the classification accuracy, multiple sets of Fourier coefficients can be used in classifying the shape. The descriptor for the best normalization above is used. In addition, if one of the multiple rotations has a value of its criterion that is within 95% of the best, this normalization is also used. To reduce the possibilities that the wrong coefficient is used in the normalization, the third largest coefficient is also used if it's magnitude is within 95% of the second largest coefficient. Again, the next best normalization based on this third largest coefficient is also used if the criterion is 95% of the best normalization based on this new coefficient. So in all there can be as many as four sets of coefficients for each unknown shape. For the library features only the best normalization is used. One of the four possible normalizations is likely to match the proper template even when some noise is present.

1.4 Fourier Descriptor Feature Vector

The feature vector was formed by calculating and normalizing the coefficients. After normalization, the $\tilde{c_0}$ and $\tilde{c_1}$ components did not carry any shape information and were dropped from the feature vector. The feature vector was then formed by listing the real and imaginary parts of the remaining components as

$$\vec{f} = (\Re\{c_{-1}\},\ \Im\{c_{-1}\}, \Re\{c_2\}, \Im\{c_2\},$$
$$\Re\{c_{-2}\}, \Im\{c_{-2}\}, \ldots, \Re\{c_{N/2}\}, \Im\{c_{N/2}\})^T.$$

This is the full feature vector. The 32 coefficients, c_{-16} to c_{15}, are used for the experiments.

2 Walsh Points

2.1 Introduction

The Walsh functions have often been used as the basis for functional approximation because the piecewise constant form leads to efficient computation [3]. Also, since the basis functions are non-zero over the entire interval of definition, this is a complete global shape analysis method. Instead of the transform coefficients, this method uses the basis functions to formulate the computation of *Walsh points* for the shape feature set. The Walsh points are a collection of points, $x_k + iy_k$, $k = 0, 1, 2, \ldots, 2^{m-1}$, that have a property derived from the fact that the Walsh functions were used in formulating their calculation. The projections $x(t)$ and $y(t)$ are approximated to within a prespecified degree of accuracy by a piecewise constant function, a truncated Walsh series.

The Walsh functions, $W_i(t)$, are products of the Rademacher functions, $r_i(t)$. The Walsh functions for a complete orthogonal basis set complete among square integrable functions on the interval $[0, 1]$. The Walsh functions are defined as

$$\begin{aligned} W_0(t) &= 1. \\ W_n(t) &= r_{n_1+1}(t) r_{n_2+1} \cdot \ldots \cdot r_{n_p+1}(t), \end{aligned}$$

where $n \geq 1$ is expressed in binary as

$$n = 2^{n_1} + 2^{n_2} + \ldots + 2^{n_p}$$

and the integers n_i are ordered such that

$$n_1 < n_2 < \ldots < n_p.$$

The Rademacher functions are

$$\begin{aligned} r_0(t) &= 1 \\ r_1(t) &= \begin{cases} +1, t \in [0, \frac{1}{2}) \\ -1, t \in [\frac{1}{2}, 1) \end{cases} \\ r_1(t+1) &= r_1(t) \\ r_{k+1}(t) &= r_1(2^k t), k = 0, 1, 2, \ldots \end{aligned}$$

In the following only the x projection of the boundary calculation will be discussed. The same results follow for the y projection.

The Walsh series for $x(t)$ is

$$x(t) = \sum_{n=0}^{\infty} \alpha_i W_i(t/T)$$

where

$$\alpha_i = \int_0^T x(t)W_i(t/T)dt = T\int_0^1 x(Tt)W_i(t)dt$$
$$T = \{\text{total arc length along } \gamma(t) = x(t) + iy(t)\}.$$

Each α_i is made up of the area under the portions of $x(t)$ added and subtracted together. In approximating $x(t)$, only a finite number of terms in the Walsh series are retained - say the first 2^m. If the series is truncated to a finite number of terms, there is an approximation error. It would be expected that as $N = 2^m$ increases, the closer $x_{2^m}(t)$ approximates $x(t)$.

Let

$$x_{2^m}(t) = \sum_{k=0}^{2^m-1} \alpha_k W_k(t/T).$$

Since the function $x_{2^m}(t)$ (or $y_{2^m}(t)$) describes a piecewise constant function, there are 2^m discrete points, $x_k + iy_k$, that are the sum of the heights of the Walsh functions over each interval, $[\frac{k}{2_m}T,\ \frac{k+1}{2^m}T)$, $k = 0, 1, \ldots, 2^m - 1$, which approximate the x (or y) projection of $\gamma(t) = x(t) + iy(t)$.

It was mentioned earlier that the Walsh functions allow for an efficient computation of the function $x_{2^m}(t)$ (or $y_{2^m}(t)$.) The i^{th} Walsh function can be written as

$$W_i(t) = \sum_{k=0}^{2^m-1} \gamma_{ik}\chi_{[\frac{k}{2^m},\frac{k+1}{2^m})}(t),$$

where

$$\gamma_{ik} = W_i(t) = \pm 1,\ \ t \in [\frac{k}{2^m}, \frac{k+1}{2^m}).$$

The matrix of γ_{ik}, $H = [H_{ik}] = [\gamma_{ik}]$, is the Hadamard matrix with 2^m rows and columns.

The i^{th} coefficient of the Walsh series is

$$\alpha_i = T\int_0^1 x(Tt)W_i(t)dt = T\sum_{k=0}^{2^m-1}\gamma_{ik}\int_0^1 x(Tt)\chi_{[\frac{k}{2^m},\frac{k+1}{2^m})}dt$$

$$\begin{aligned}\alpha_i &= \sum_{k=0}^{2^m-1}\gamma_{ik}\int_{\frac{kT}{2^m}}^{\frac{(k+1)T}{2^m}} x(t)dt \\ \alpha_i &= \sum_{k=0}^{2^m-1}\gamma_{ik}A_{xk}, \qquad (3)\end{aligned}$$

where

$$A_{xk} = T\int_{\frac{k}{2^m}}^{\frac{k+1}{2^m}} x(Tt)dt = \int_{\frac{kT}{2^m}}^{\frac{(k+1)T}{2^m}} x(t)dt.$$

A_{xk} is the area under $x(t)$ over the basic interval $[\frac{kT}{2^m}, \frac{(k+1)T}{2^m})$. By letting

$$\vec{A_x} = \begin{pmatrix} A_{0x} \\ A_{1x} \\ \vdots \\ A_{2^m-1x} \end{pmatrix}$$

be the *Area* vector and

$$\vec{\alpha_x} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^m-1} \end{pmatrix}$$

be the *sequency* vector, we can write

$$\vec{\alpha_x} = H\vec{A_x}.$$

The Truncated Walsh series is

$$\begin{aligned} x_{2^m}(t) &= \sum_{i=0}^{2^m-1} \alpha_i W_i(t/T) \\ &= \sum_{i=0}^{2^m-1} \alpha_i \left(\sum_{k=0}^{2^m-1} \gamma_{ik}\chi_{[\frac{k}{2^m},\frac{k+1}{2^m})}(t/T) \right) \\ &= \sum_{k=0}^{2^m-1} \left(\sum_{i=0}^{2^m-1} \alpha_i\gamma_{ik} \right) \chi_{[\frac{k}{2^m},\frac{k+1}{2^m})}(t/T) \\ &= \sum_{k=0}^{2^m-1} c_k\chi_{[\frac{k}{2^m},\frac{k+1}{2^m})}(t/T), \end{aligned}$$

where

$$c_k = \sum_{i=0}^{2^m-1} \gamma_{ik}\alpha_i. \tag{4}$$

The constants c_k can be found by first forming the vector $\vec{x_{2^m}}$ as

$$\vec{x_{2^m}} = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2^m-1} \end{pmatrix}$$

and noticing that as in equation (1) that equation (2) can be written as

$$\vec{x_{2^m}} = H^T \vec{\alpha_x}$$

where H^T is the transpose of the Hadamard matrix, H.

Since H is symmetric $H = H^T$ and

$$\vec{x_{2^m}} = H^2 \vec{A_x}.$$

But because the Walsh functions are orthogonal, it follows that $H^2 = 2^m$. Hence,

$$\vec{x_{2^m}} = 2^m \vec{A_x}.$$

So, the truncated Walsh series gives a piecewise constant function whose heights are proportional to the area under the projection on each basic interval, i.e.

$$x_{2^m}(t) = 2^m \sum_{k=0}^{2^m-1} A_{kx} \chi_{[\frac{kT}{2^m}, \frac{(k+1)T}{2^m})}(t).$$

2.2 Normalization

In order to compare the Walsh points feature vector of an unknown to a library prototype, normalization with respect to translation, scale, rotation, and shift in starting point must be accomplished.

The traced contour from the image grid is stored using Freeman chain code links. When the shape features are to be extracted, the chain code is converted to a complex vector $\vec{\gamma} = \vec{x} + i\vec{y}$. The increments in the arc length for the piecewise linear segments connecting the grid points are calculated. The c_0, c_1, and c_{-1} Fourier coefficients of the boundary are calculated to provide the global information necessary for translation, scale, and rotation.

The translation is $-c_0$. The scale factor $s = \mid c_1 \mid$. The rotation angle is $\psi = -\frac{\angle c_1 + \angle c_{-1}}{2}$. The complex vector $\vec{\gamma} = \begin{pmatrix} \gamma_0 \\ \vdots \\ \gamma_k \end{pmatrix}$ is normalized to $\vec{\acute{\gamma}}$, where the i^{th} component is

$$\acute{\gamma}_i = \frac{1}{s}(\gamma_i - c_0)e^{i\psi}, \text{ and } \acute{T} = \frac{T}{s}.$$

This normalization moves the center of the shape to the origin. It scales and rotates the shape so that the major axis corresponding to the fundamental ellipse is along the x-axis and has a length of 2. The starting point is moved by finding the k^{th} point, γ_k, which is closest to one of the two points where the fundamental ellipse crosses the x-axis. An additional 180° rotation is applied if necessary to place $\acute{\gamma}_k$ in the right half plane. So, the normalized Walsh points form the vector

$$\underline{\vec{\gamma}} = \begin{pmatrix} \acute{\gamma}_k \\ \vdots \\ \gamma_{\acute{K}-1} \\ \acute{\gamma}_0 \\ \vdots \\ \gamma_{\acute{k}-1} \end{pmatrix} = \underset{k}{\text{shift}} \; \vec{\acute{\gamma}}.$$

or $\underline{\vec{\gamma}} = \underset{k}{\text{shift}} \; \vec{\acute{\gamma}} \cdot e^{i\frac{\pi}{2}}$. Then the Walsh points are calculated by computing the Area vector and multiplying by 2^m.

2.3 Walsh Points Feature Vector

The feature vector used in the experiments was formed by simply listing the Fourier points real and imaginary parts in order, i.e.

$$\underline{f} = (x_0, y_0, x_1, y_1, \ldots, x_{K-1}, y_{K-1})^T$$

For the experiments the full feature vector consisted of 32 Walsh points.

3 Shape Library Generation

The focus of this study is to determine the sensitivity of the shape recognition algorithms to the density of library entries and to imaging noise. The algorithms are examined for their performance for correct helicopter fuselage type recognition and orientation determination.

A list of shape features are extracted from the outer contour of the silhouette of a helicopter fuselage. These features are derived from computer models of the helicopter fuselage. A computer program is used to obtain a three-dimensional model of the solid object. Additional computer graphics programs are used to rotate the internal model and produce a computer generated image of the solid object. Therefore, for each shape that the algorithms are to recognize, it is necessary to enter into the computer its three-dimensional description. A three-dimension solid modelling program, Mac3D [11], generates the solid models. This program allows the user to create a three-dimensional model by specifying the size and orientation of simple three-dimensional objects. These objects include spheres, cones, parallel-pipeds, tori, and surfaces of revolution. Very complex objects can be modelled by combining these simple shapes. A good approximation to the real world object shape can be obtained in this manner.

The helicopter rotors blades are not included in the modelling of the helicopter fuselage. This is because experiments in segmenting the helicopter fuselage have indicated that the rotor blades are in fact difficult to extract from the image. This topic is discussed in detail in Section 7.

The three-dimensional specifications for three military helicopters are entered into the solid modelling program for this study. The descriptions are taken from Jane's All the World's Aircraft [9]. The helicopters choosen are: the Bell Model 205 UH-1 Iroquis, the McDonnell Douglas 500MD Defender, and the Sikorsky UH-60A Black Hawk.

Once the three-dimensional object description is available, the image of the helicopter in any arbitrary orientation can be obtained. The object rotation transformations are discussed in Section X. Before the classification and orientation experiments are performed, a library of views is generated for each helicopter type. For each library for the three types of helicopters, the following procedure is carried out. For each view to be stored in the library, the object is rotated by the specified rotation about the x and y axis,θ_x and θ_y, respectively. The object coordinate points are then projected onto the image plane. The image formed is the silhouette of the helicopter fuselage viewed in the specified orientation. The the contour of this silhouette is then traced using a 8-direction chain-code. The chain-codes for each stored view are collected into the library for that helicopter type. The angle orientation information is retained in a special data record with each view for latter use. The chain-code libraries are then processed using one of the two shape recognition algorithms to extract the shape features. The normalized shape features for each helicopter type are stored together in a feature vector library.

Once the libraries containing the shape features for each helicopter have been generated, experiments can be performed. The performance of each helicopter shape is determined by classifying labelled "unknown"s from each helicopter type viewed at random orientations. Shape feature vectors are classified using a nearest neighbor template matching. The normalized shape feature vector is compared to every feature vector in each of the three libraries. The unknown is assigned to the helicopter type and orientation corresponding to the closest match. The distance measure used for the template matching is the sum of the squared differences between corresponding feature vector elements, i.e.

$$d_{L_j} = \sum_{i=1}^{K} (\hat{f}_i^u - \hat{f}_i^{L_j})^2$$

where $\hat{f}_i^{L_j}$ is the i^{th} element in the j^{th} feature vector of the L^{th} helicopter normalized shape feature vector and $\hat{f}_i^u$ is the i^{th} element of the unknown normalized shape feature vector.

Libraries for each helicopter type are generated at various angle sampling densities. The views that make up each library are taken from equally spaced values in the x and y orientation angles. The x and y angle rotations range from 0 to $-\pi$ and $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, respectively. For each helicopter type, four different libraries are generated for the purpose of determining the sensitivity of classification and orientation accuracy to the density of the library . The four libraries for each helicopter type contain either 5x3, 7x5, 9x7, or 11x9 views.

4 Experimental Procedure

Two groups of experiments are performed to determine the performance of the two shape recognition algorithms as measured by helicopter fuselage type classification and orientation angle accuracy. The purpose of the first group experiment is to determine the sensitivity of the algorithm performance as a function of the density of reference library entries. The second group of experiments is to determine the sensitivity to imaging noise.

For the first group of experiments, libraries for each helicopter type are generated at various angle sampling densities. The views that make up each library are taken from equally spaced values in the x and y orientation angles. The x and y angle rotations range from 0 to $-\pi$ and $-\frac{\pi}{2}$ to $\frac{\pi}{2}$, respectively. For each helicopter type, four different libraries are generated. The four libraries for each helicopter type contain either 5x3, 7x5, 9x7, or 11x9 views corresponding to orientation angle sampling intervals of 36° × 60°, 25.7° × 36°, 20° × 25.7°, and 16.4° × 20°. The contours for each of the three helicopter types at each of the four library sampling densities are shown in Figures 1 - 12. In the libraries shown in the figures, the variation in θ_y are horizontal and the variations in θ_x are vertical. The $(\theta_x, \theta_y) = (0, 0)$ is at the top center.

For each classification experiment, the unknowns are compared to a library for each helicopter type, all at the same library density. In each experiment, 63 random views of each of the three helicopter types are classified. The classification accuracy and statistics on the angle orientation errors are computed and tabulated. This procedure is carried out using for both the Fourier descriptors and the Walsh points. A small sample of the "unknowns" are shown in Figures 14 - 16.

The purpose of the second set of experiments is to examine the effect of imaging noise on classification accuracy and orientation determination of a helicopter fuselage. In addition, the experiments are performed with the feature vector shape libraries at each of the four sampling densities. As before, the outer contours for 63 random views of each helicopter type are generated using the solid modelling program followed by object coordinate rotation and projection. The outer contour of the rotated and projected surface edges of the solid object are traced. Next, the interior of the contour

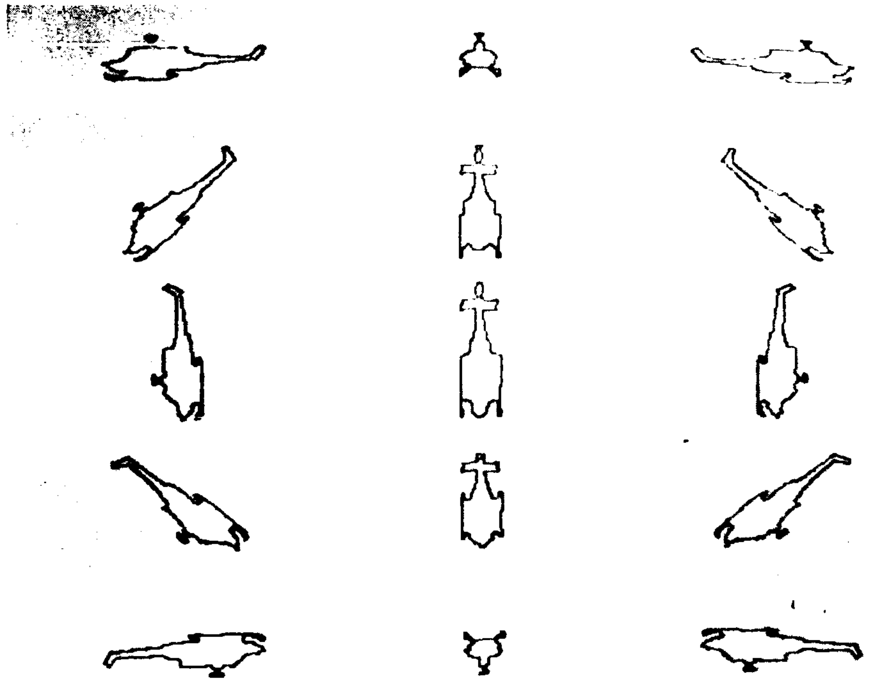


Figure 1: Bell Model 205 UH-1 Iroquois library of 5x3 (15) views.

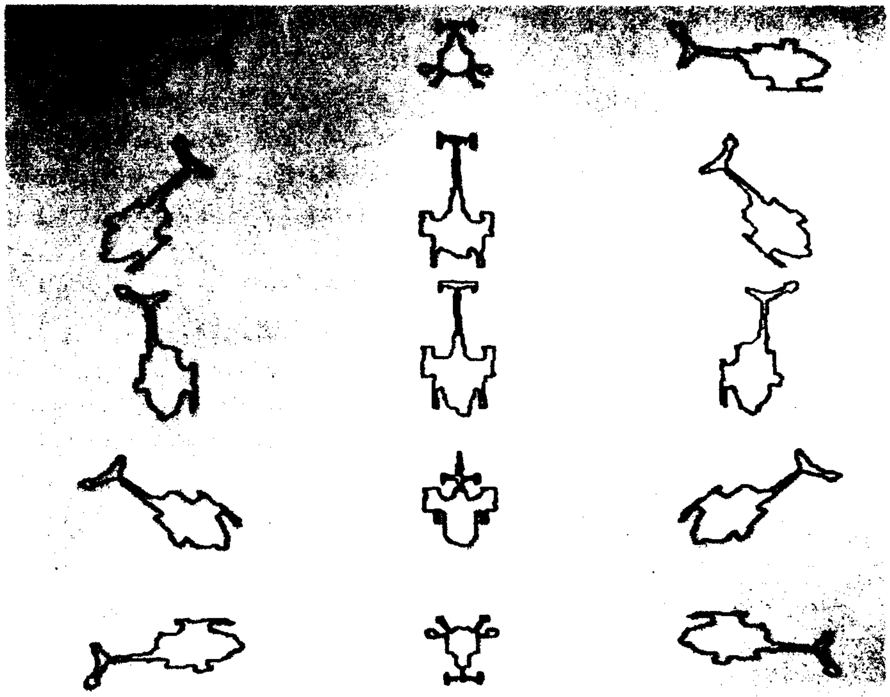


Figure 2: McDonnell Douglas 500MD Defender library of 5x3 (15) views.

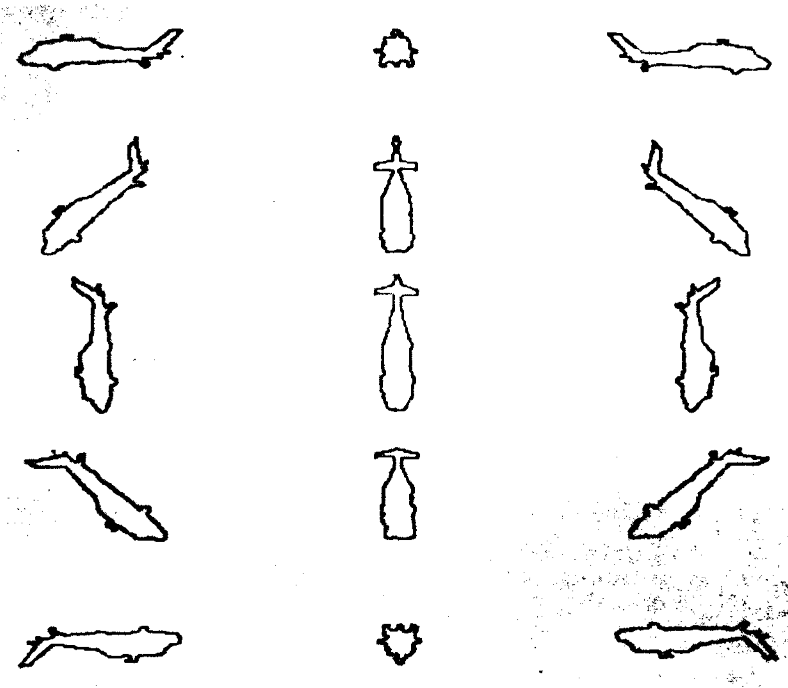


Figure 3: Sikorsky UH60A Black Hawk library of 5x3 (15) views.

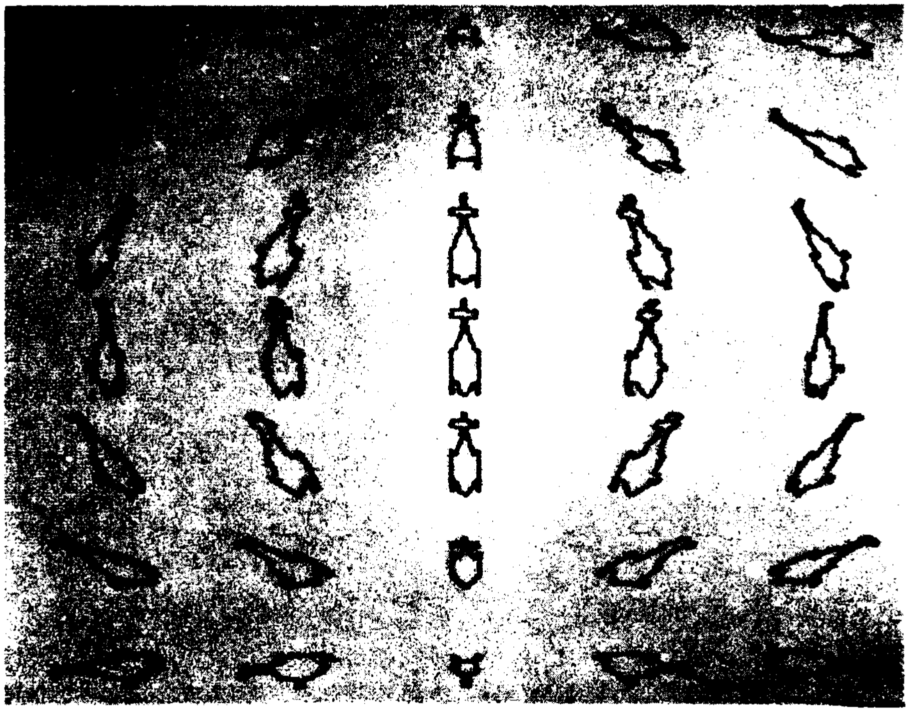


Figure 4: Bell Model 205 UH-1 Iroquois library of 7x5 (35) views.

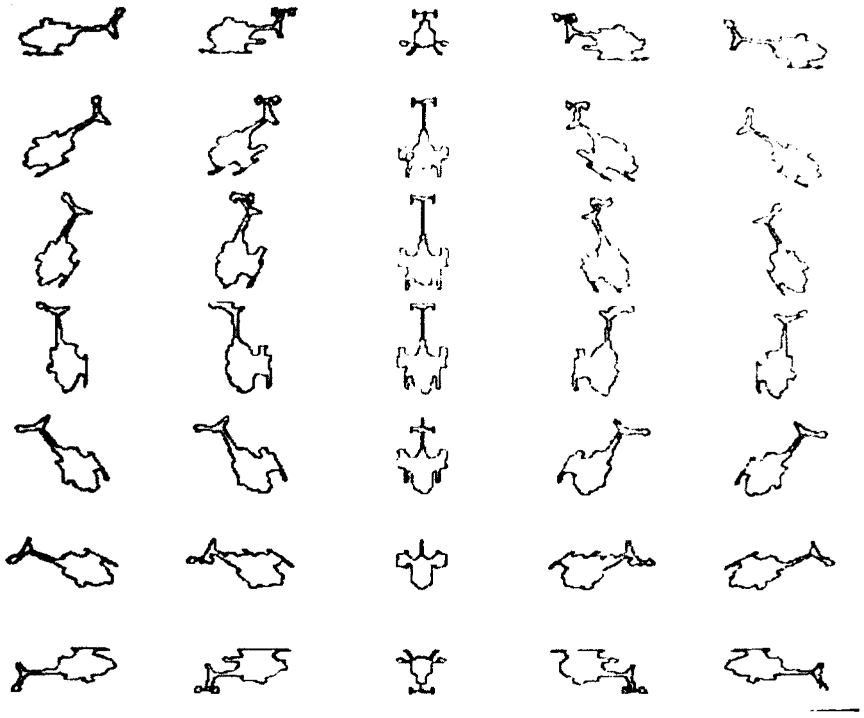


Figure 5: McDonnell Douglas 500MD Defender library of 7x5 (35) views.

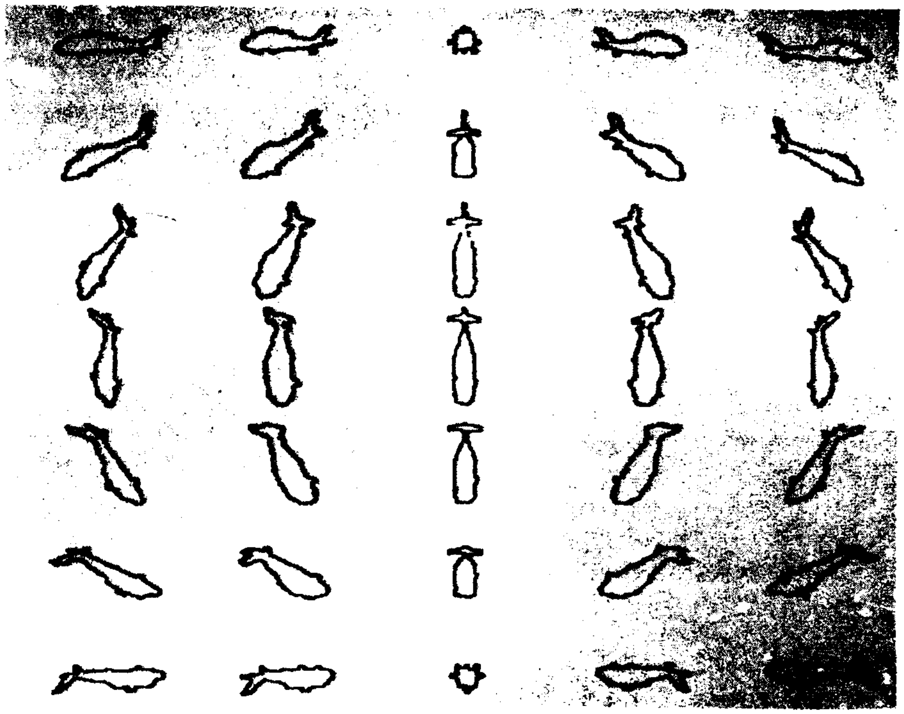


Figure 6: Sikorsky UH60A Black Hawk library of 7x5 (35) views.

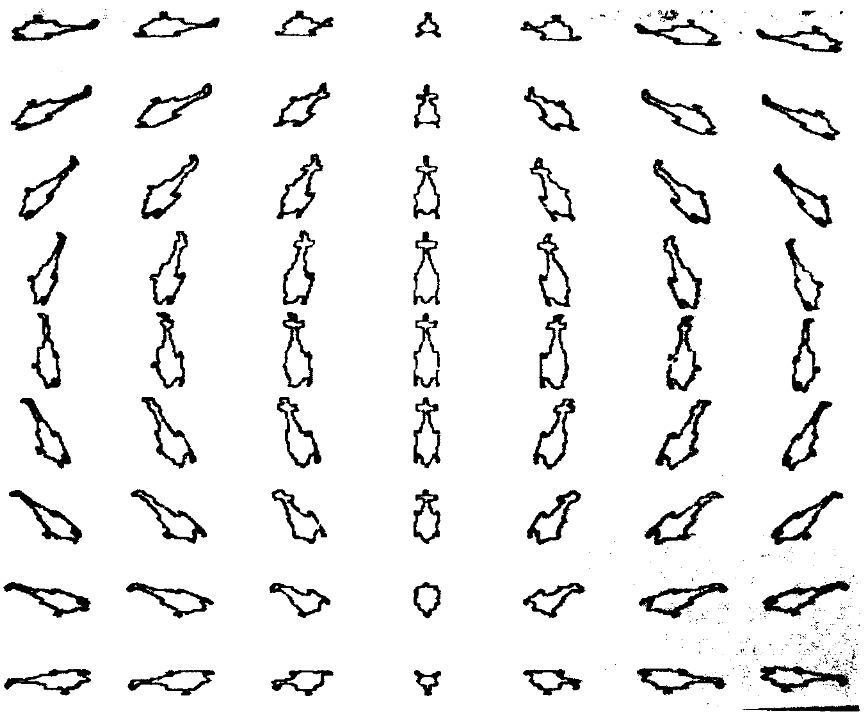


Figure 7: Bell Model 205 UH-1 Iroquois library of 9x7 (63) views.

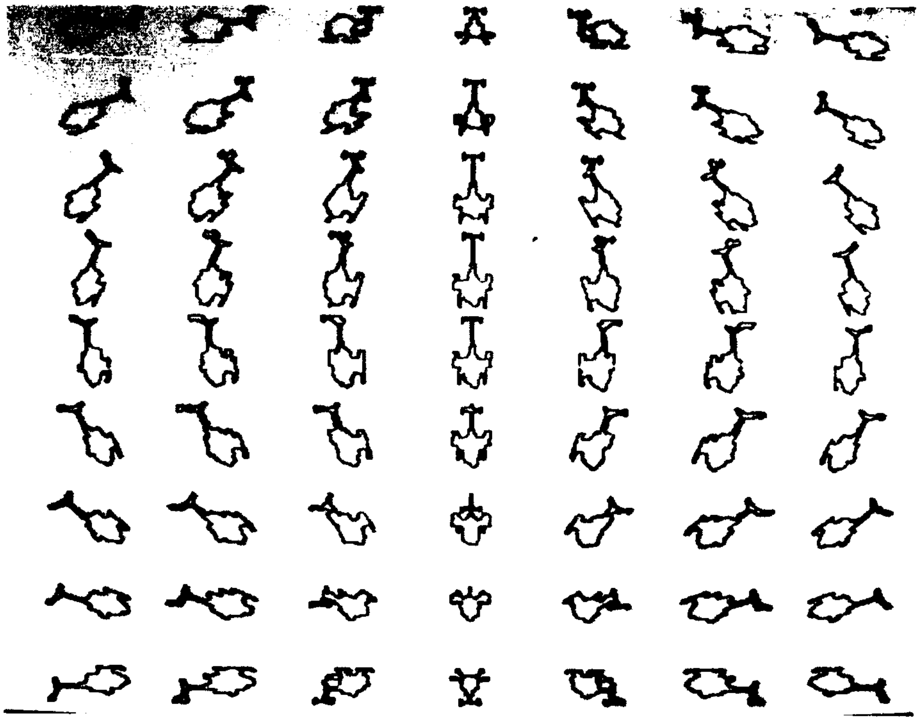


Figure 8: McDonnell Douglas 500MD Defender library of 9x7 (63) views.

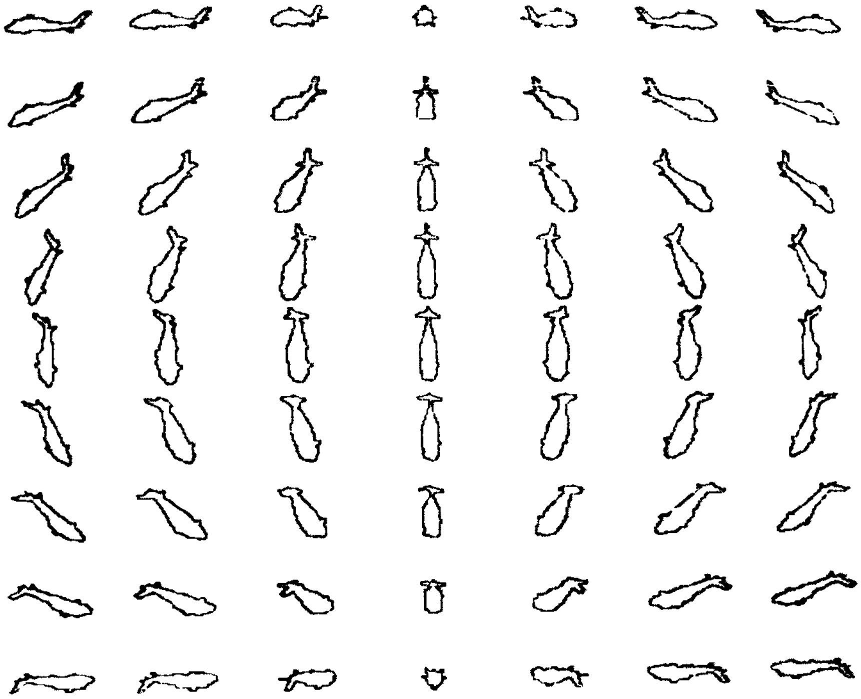
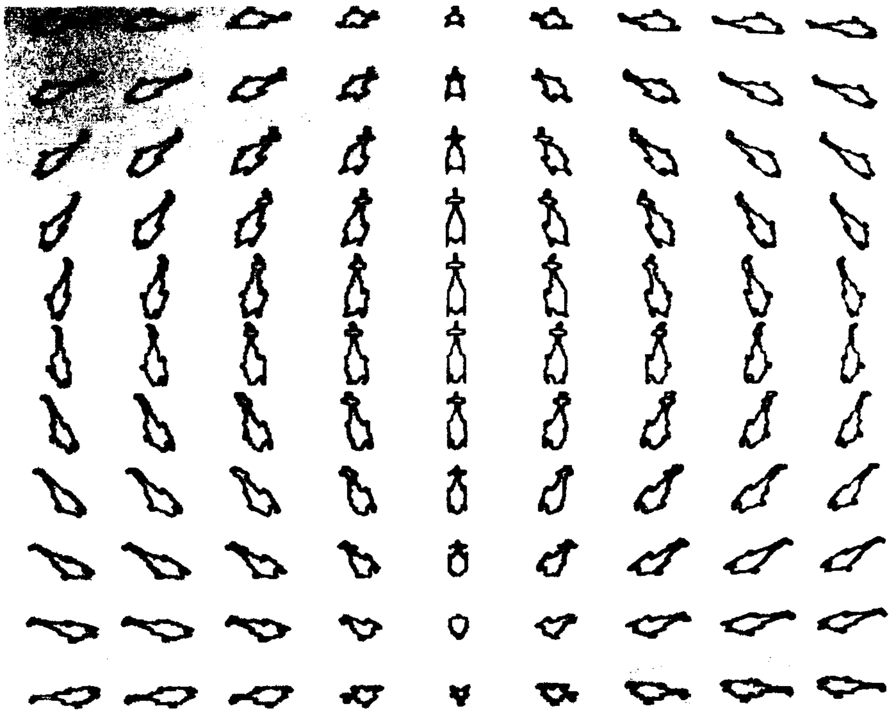


Figure 9: Sikorsky UH60A Black Hawk library of 9x7 (63) views.



Fig'ıre 10: Bell Model 205 UH-1 Iroquois library of 11x9 (99) views.

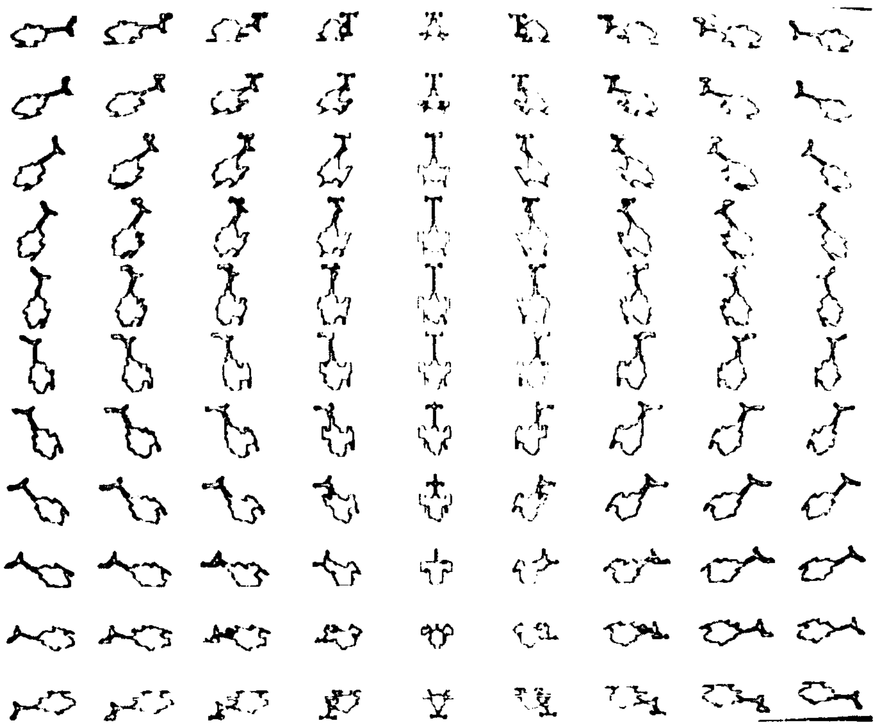


Figure 11: McDonnell Douglas 500MD Defender library of 11x9 (99) views.

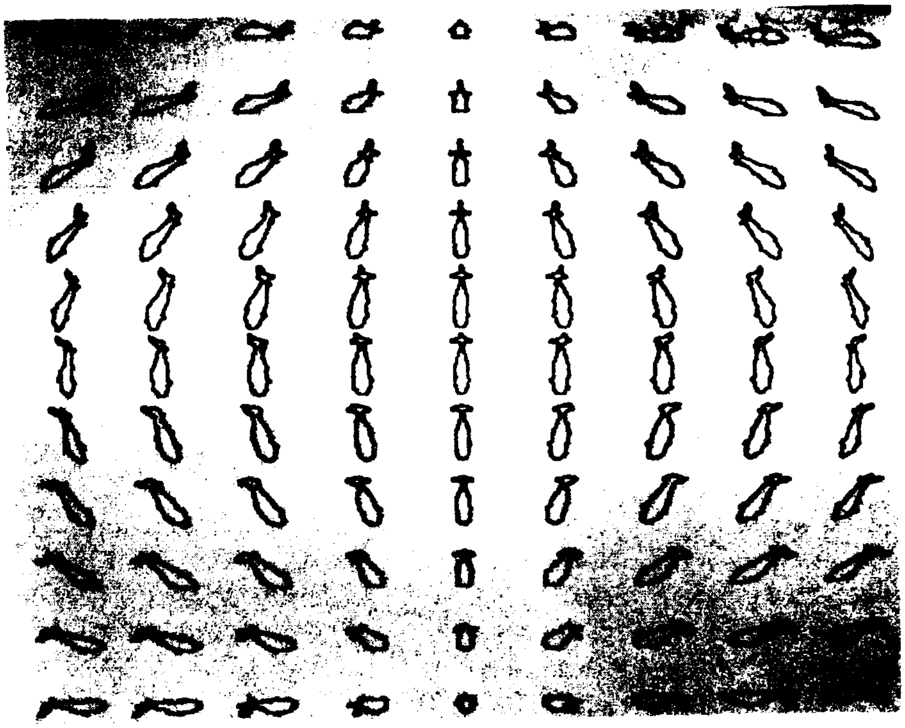


Figure 12: Sikorsky UH60A Black Hawk library of 11x9 (99) views.

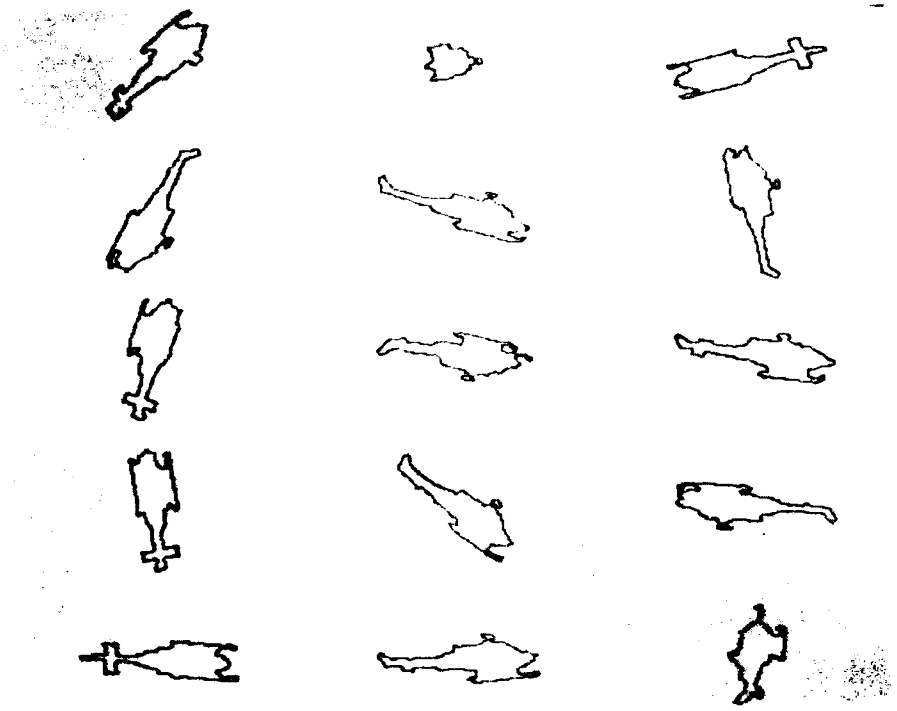


Figure 13: Example contours for 15 "unknowns"- Bell Model 205.

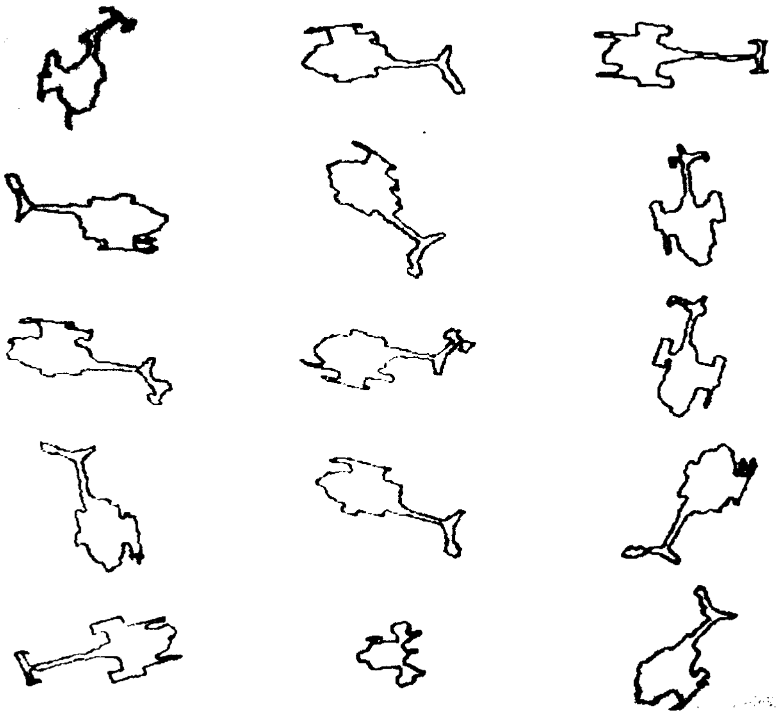


Figure 14: Example contours for 15 "unknows" - McDonnell Douglas 500MD.

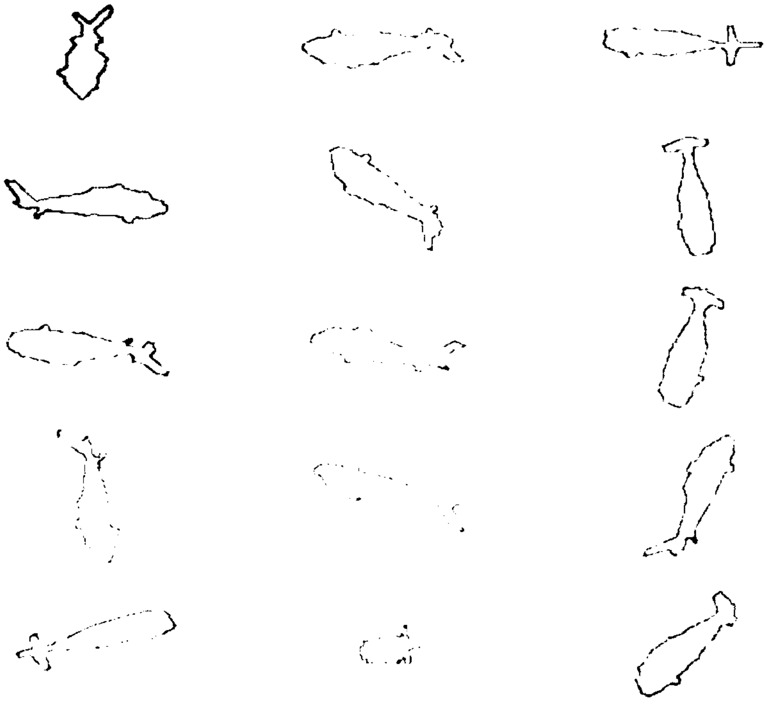


Figure 15: Example contours for 15 “unknowns” - Sikorsky UH60A.

is filled with a level of 191 and the outside area (background) the value 64. White Gaussian noise is then added to simulate image noise. The image is then thresholded at a level of 110. The outer contours in the image are then traced and the longest contour retained to describe the noisy, helicopter fuselage silhouette, outer contour. Two additional sets of labeled "unknown" helicopter contours are generated in this manner. Each set corresponds to a object intensity level above background to noise variance signal-to-noise ratio, i.e.

$$\mathrm{SNR} = 20\log(\frac{\Delta}{\sigma}),$$

where Δ = (object gray level intensity - background grey-level intensity) and σ = the standard deviation of the noise process. The classification and orientation accuracy experiments are then performed using unknowns at the three signal-to-noise ratios, infinity, 10dB, and 3 dB corresponding to noise σ of 0, 20.238, and 45.310, respectively. Examples of the images and contours at each signal-to-noise ratio for the Bell MD500 helicopter are shown in Figures 16 - 21.

5 Recognition Accuracy and Orientation Determination Results

The ability of the shape recognition algorithms to assign the "unknown" shape contour to the correct helicopter fuselage *type* are tabulated in Table 2 and shown in Figure 22. The plot and table show the classification accuracy under all the tested experimental conditions including variation in library density and image signal-to-noise ratio. Under all conditions the Fourier descriptor shape recognition algorithm out performs the Walsh points recognition algorithm. The margin is always at least and additional 12%. The performance of both algorithms degrade significantly when the number of library entries falls below 35 (7x5). With a moderate decrease in the signal-to-noise ratio, little degradation in classification accuracy occurs. However, when the signal-to-noise ratio is decreased to only 3dB, the Fourier descriptor accuracy drops some 8%. The Walsh points algorithm

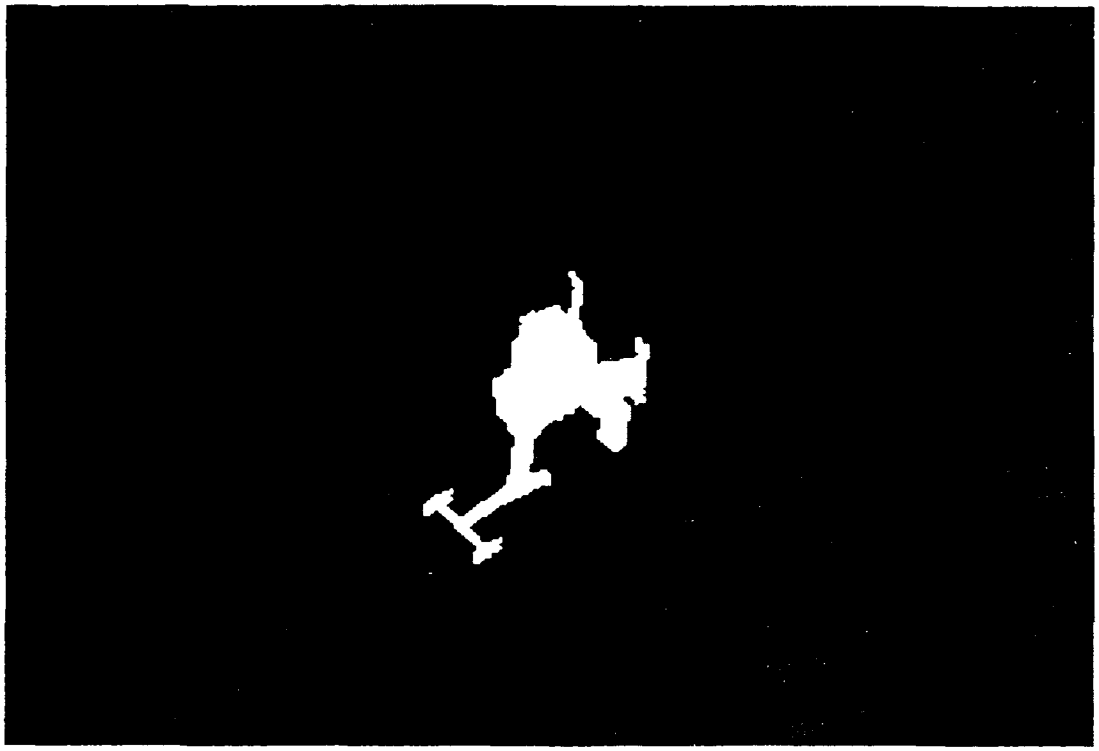


Figure 16: Bell MD500 shape silhouette image at SNR = ∞

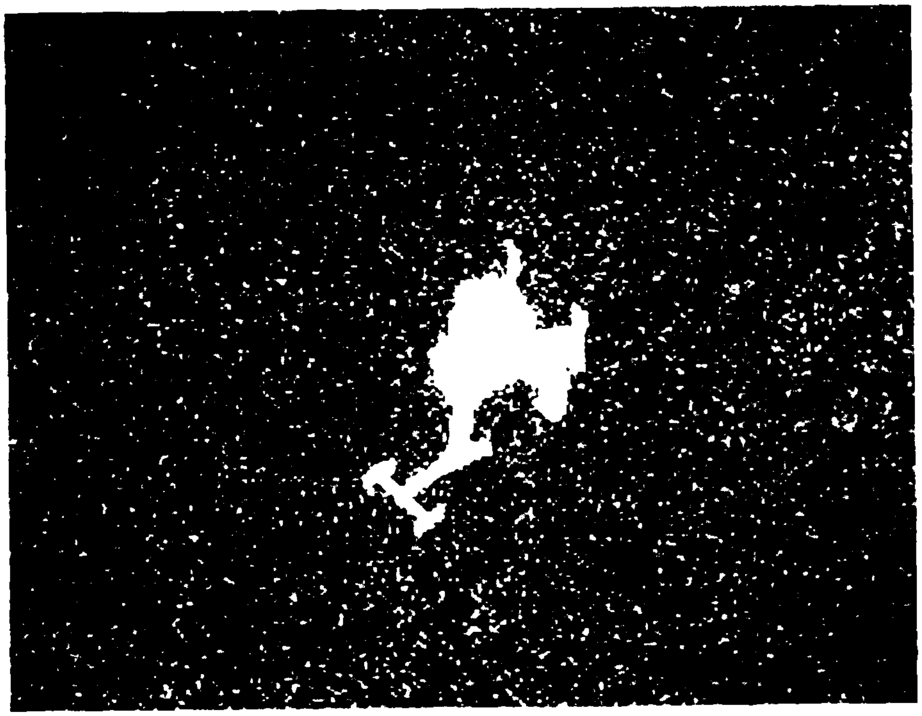


Figure 17: Bell MD500 silhouette and imaging noise at SNR = 10dB.

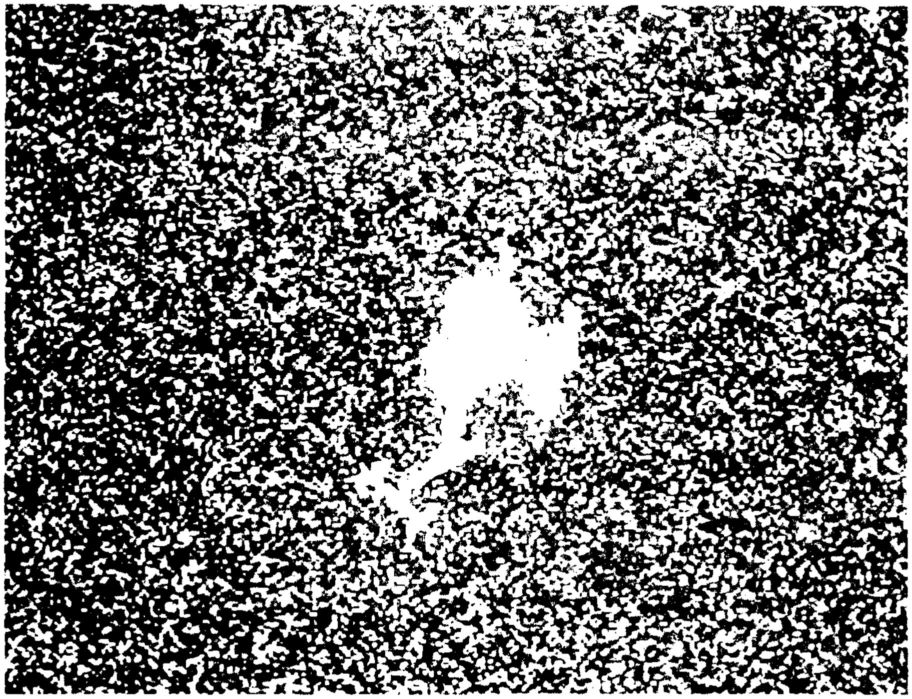


Figure 18: Bell MD500 silhouette and imaging noise at SNR = 3dB.

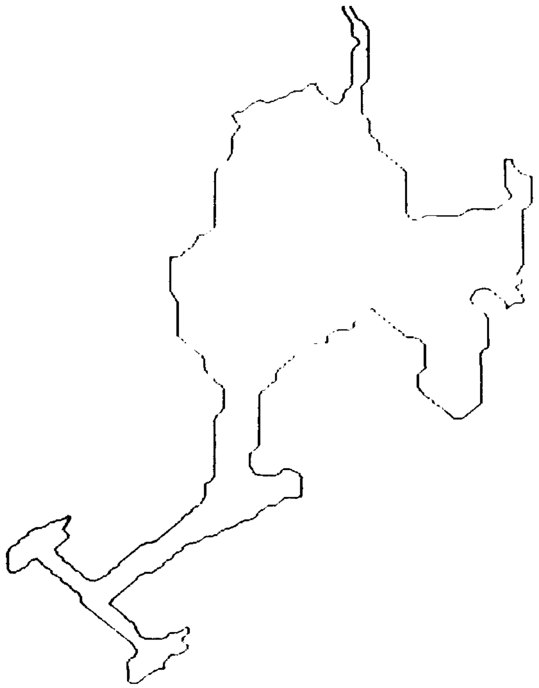


Figure 19: Bell MD500 contour with no imaging noise.

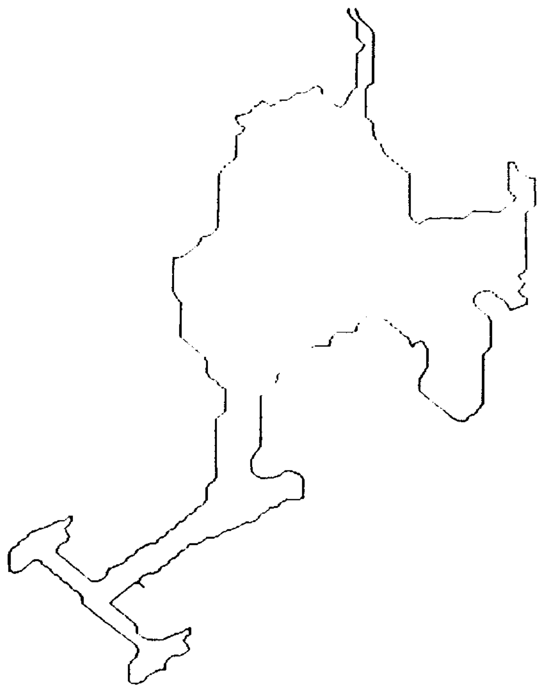


Figure 20: Bell MD500 noisy contour at SNR = 10dB.

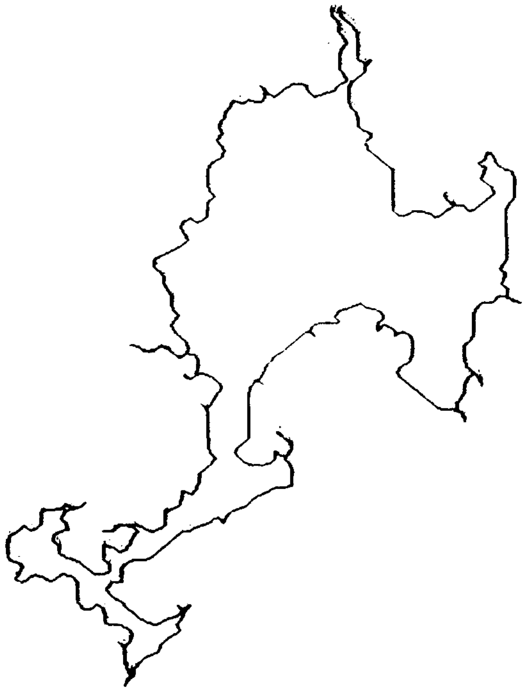


Figure 21: Bell MD500 noisy contour at SNR = 3dB.

is more sensitive and drops by at least 13% when the signal-to-noise ratio decreases from 10 dB to 3 dB. The sensitivity curves indicate a non-linear effect. The orientation normalization used in both algorithms is a non-linear decision process. This would explain this non-linear phenomenon. At a signal-to-noise ratio of 3 dB, the effects of partial contours due to segmentation errors also just begins to occur. Under such conditions, both the performance of Fourier and Walsh methods deteriorates rapidly [6].

Figures 23 thru 26 and Tables 3 and 4 show the sensitivity of the algorithms to the experimental conditions as measured by orientation angle error. Plots for the mean and median x and y angle errors are shown. Again, under almost all conditions, the Fourier descriptor method out performs the Walsh points method. The angle errors for the Walsh method being approximately twice as large as those for the Fourier method. For the Fourier descriptors, the y angle error is approximately one-half the library y angle sampling interval. This is the best that is likely to be obtained for any shape method without employing some interpolation between library views. For the Walsh points method, the y angle errors are twice that of the Fourier method.

For the Fourier descriptor method, the mean errors for the x angle orientation determination is approximately two and a half times as large as the library x angle sampling interval. The mean x angle error for the Walsh points method is as much as four times the library sampling interval. The cause of the large mean x angle errors is first indicated by examining the median x angle errors. For example, the mean x angle error for the Fourier descriptor method at the 11x9 library density is 25°, but has a median error of only 8°. Inspecting the few individual cases where there is a large x angle error, shows that the same fuselage at a displacement angle of 90° is being incorrectly assigned. The number of such instances is few but the 90° discrepancy is large enough to greatly bias the sample mean error. The effect of such errors could be mitigated in an operating scenario, by smoothing the angle orientations after locking on to the target trajectory.

The median angles for the Walsh points method are also small indicating the same problem. However, the performance is already inferior to that of the Fourier method.

In summary, the overall performance of the Fourier descriptor method is superior to that of the Walsh points method. Both methods, however, exhibit large *mean* x angle errors. However, most of the error is due to a few large miss assignments in orientation that can be post-filtered by a tracking algorithm which incorporating the limits of the helicopter dynamics.

6 Rotor Orientation Determination

In this section, a scheme for segmenting and subsequently determining the helicopter rotor orientation relative to the helicopter fuselage will be discussed. If rotor orientation can be determined from the imagery in a timely fashion, it should be possible to provide a more accurate prediction of the future trajectory of the helicopter.

In pursuit of the above goal, video imagery of a maneuvering helicopter was obtained and digitized. Several algorithms were applied in an attempt to segment the rotor blade "disk" from the image. All the methods failed to produce a useable segmentation. It is this same fact, however, that makes it possible to apply the global shape methods in Sections 1 and 2 of this report. The difficulty in segmenting the rotor blades reduces the confusion in providing a complete shape of the helicopter silhouette. However, further investigation in to the acquisition of the data on the video tape indicates that an alternative should exist.

The data recorded on the video tape was acquired using a vidicon camera. This sensor type experiences image lag. Therefore, the image of the individual rotor blades were integrated temporally resulting in a single very low contrast elliptical disk of the rotor blade tip path. (For most orientations of the helicopter fuselage.) By using a different sensor type, e.g. a CCD camera, the effects of temporal averaging can be significantly reduced. Such a sensor operating at standard rate of 60 fields per second would image the rotor blade through only 0.1 of a revolution. (Assuming rotor radius of 27 feet and a tip speed of Mach 1.) Having a short integration time requires that the camera iris be open as wide as possible, thereby reducing the depth of focus. This forces the constraint that the camera already be locked on the target in a tracking mode. Subsequently, the contrast between nominal background and the rotor blades should be significantly enhanced. Making

Shape Method	Library Density	SNR (dB)	Classification Accuracy (%)
FDS	11x9	$\propto$	99.47
		10	99.47
		3	94.18
	9x7	$\propto$	97.88
		10	97.35
		3	92.06
	7x5	$\propto$	95.77
		10	95.77
		3	89.42
	5x3	$\propto$	87.36
		10	85.19
		3	84.13
WAL	11x9	$\propto$	85.19
		10	83.60
		3	73.02
	9x7	$\propto$	85.71
		10	85.19
		3	69.84
	7x5	$\propto$	78.84
		10	77.25
		3	66.14
	5x3	$\propto$	70.90
		10	68.78
		3	62.43

Table 2: Performance: Classification Accuracy

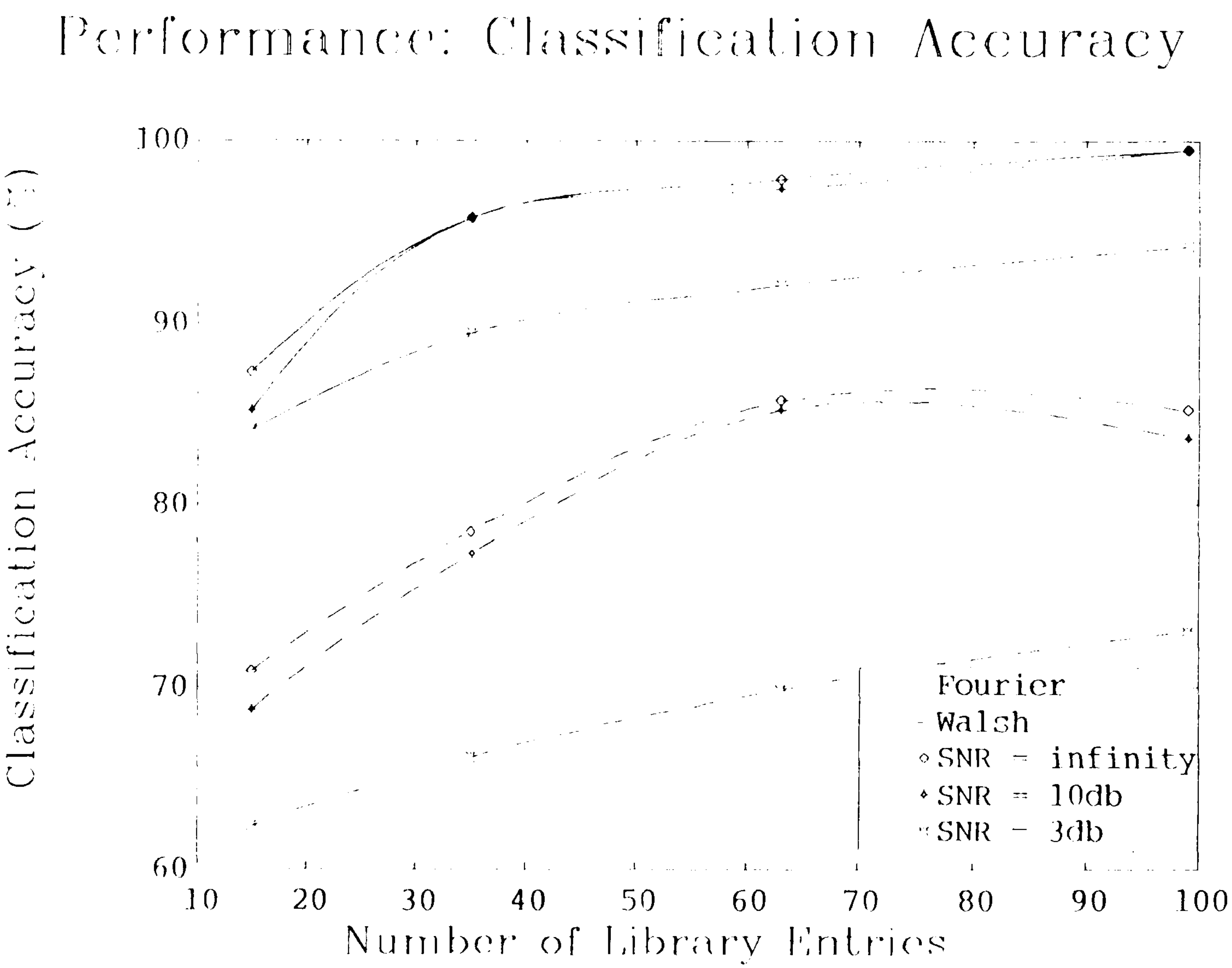


Figure 22: Performance: Classification Accuracy

Shape Method	Library Density	SNR (dB)	Mean X Angle Error(°)	Mean Y Angle Error(°)
FDS	16.5°x20°	∝	25.0	6.9
		10	27.2	7.0
		3	33.9	12.5
	20°x25.7°	∝	26.6	8.5
		10	26.1	8.6
		3	38.4	13.4
	25.7°x36°	∝	36.8	13.9
		10	38.0	14.0
		3	45.3	17.2
	36°x60°	∝	73.1	25.2
		10	74.1	25.2
		3	63.6	26.1
WAL	16.5°x20°	∝	58.4	21.0
		10	56.3	21.1
		3	62.8	24.2
	20°x25.7°	∝	57.6	19.1
		10	59.0	19.1
		3	66.4	26.1
	25.7°x36°	∝	65.2	28.2
		10	58.1	31.5
		3	60.3	33.2
	36°x60°	∝	70.5	50.4
		10	70.6	53.3
		3	67.9	55.3

Table 3: Performance: Mean Angle Error

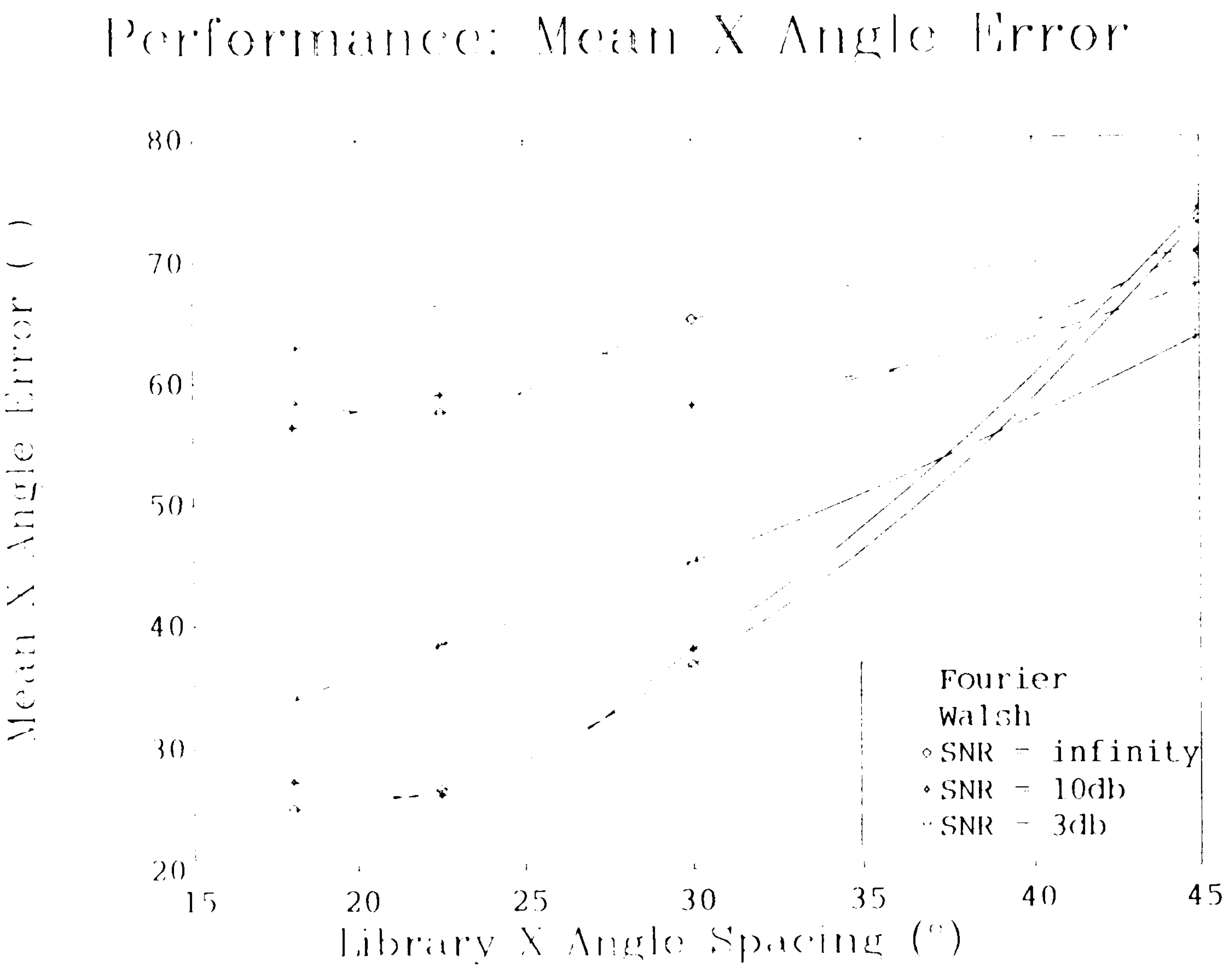


Figure 23: Performance: Mean X Angle Error

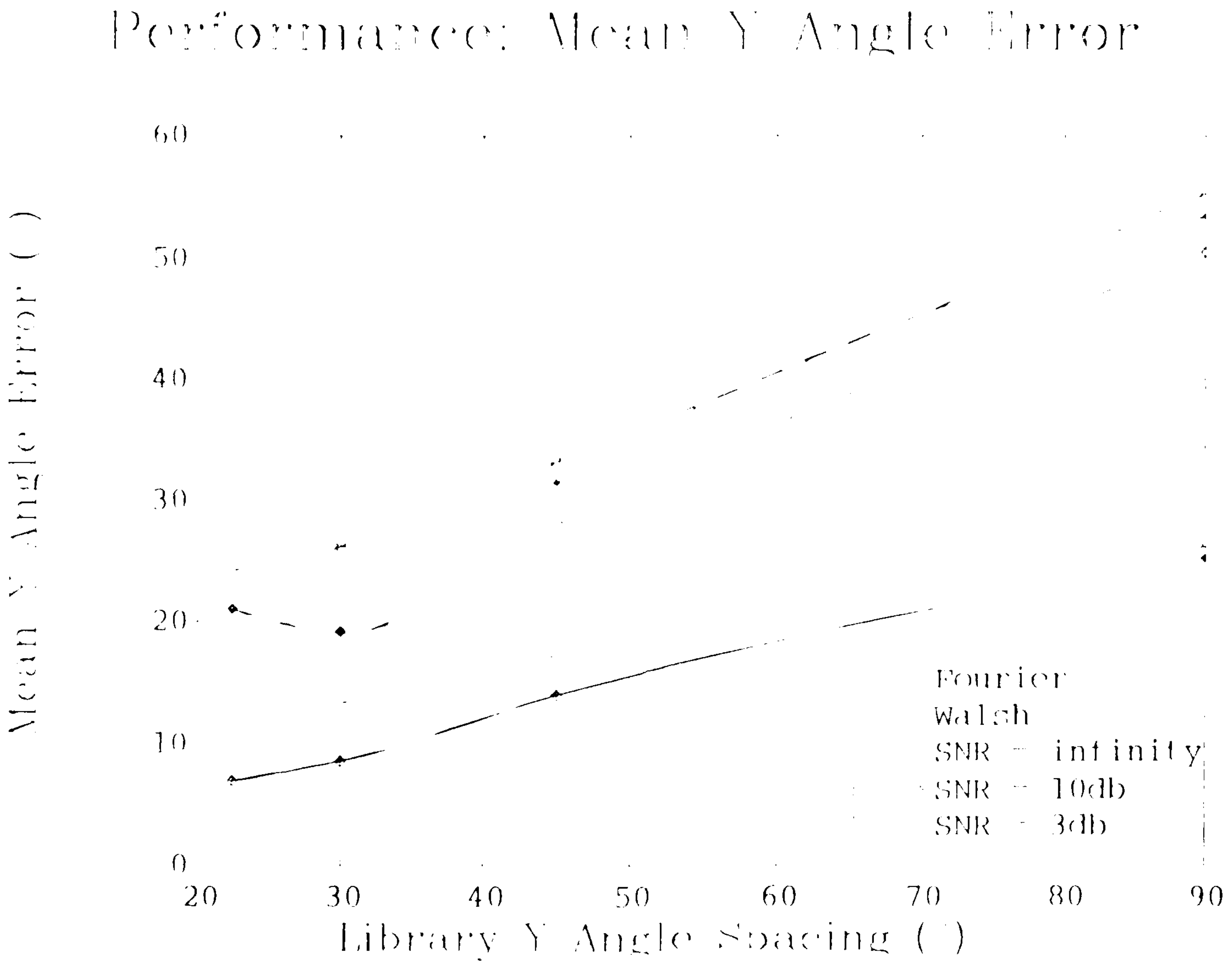


Figure 24: Performance: Mean Y Angle Error

Shape Method	Library Density	SNR (dB)	Median X Angle Error(°)	Median Y Angle Error(°)
FDS	16.5°x20°	∞	8.0	7.0
		10	9.0	7.0
		3	14.0	9.0
	20°x25.7°	∞	10.0	8.5
		10	10.0	6.0
		3	18.0	10.0
	25.7°x36°	∞	17.0	12.0
		10	17.0	12.0
		3	22.0	15.0
	36°x60°	∞	67.0	25.0
		10	68.0	25.0
		3	51.0	25.0
WAL	16.5°x20°	∞	36.0	11.0
		10	31.0	12.0
		3	44.0	13.0
	20°x25.7°	∞	31.0	12.0
		10	35.0	12.0
		3	51.0	18.0
	25.7°x36°	∞	44.0	16.0
		10	40.0	18.0
		3	47.0	26.0
	36°x60°	∞	59.0	36.0
		10	62.0	34.0
		3	59.0	38.0

Table 4: Performance: Median Angle Error

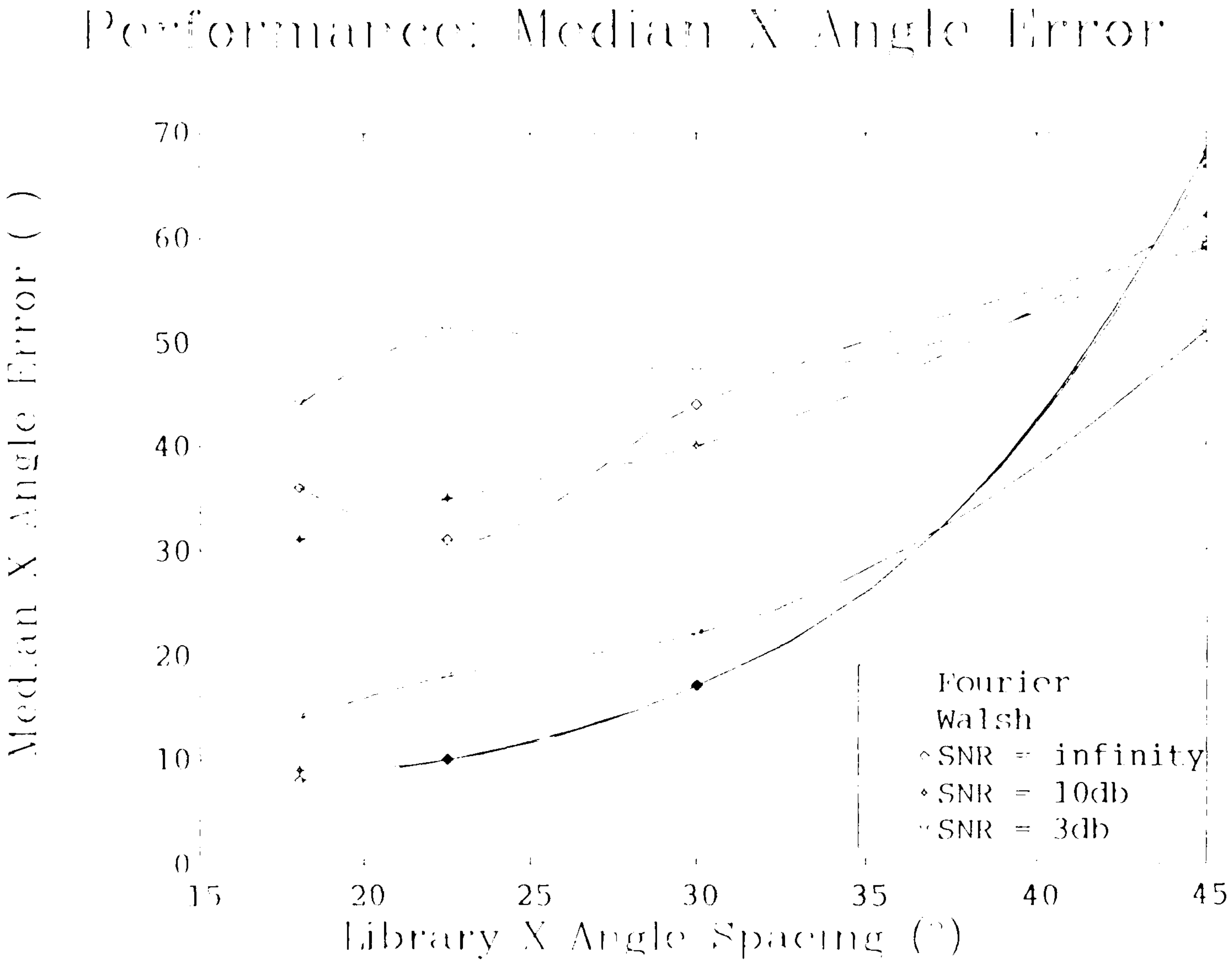


Figure 25: Performance: Median X Angle Error

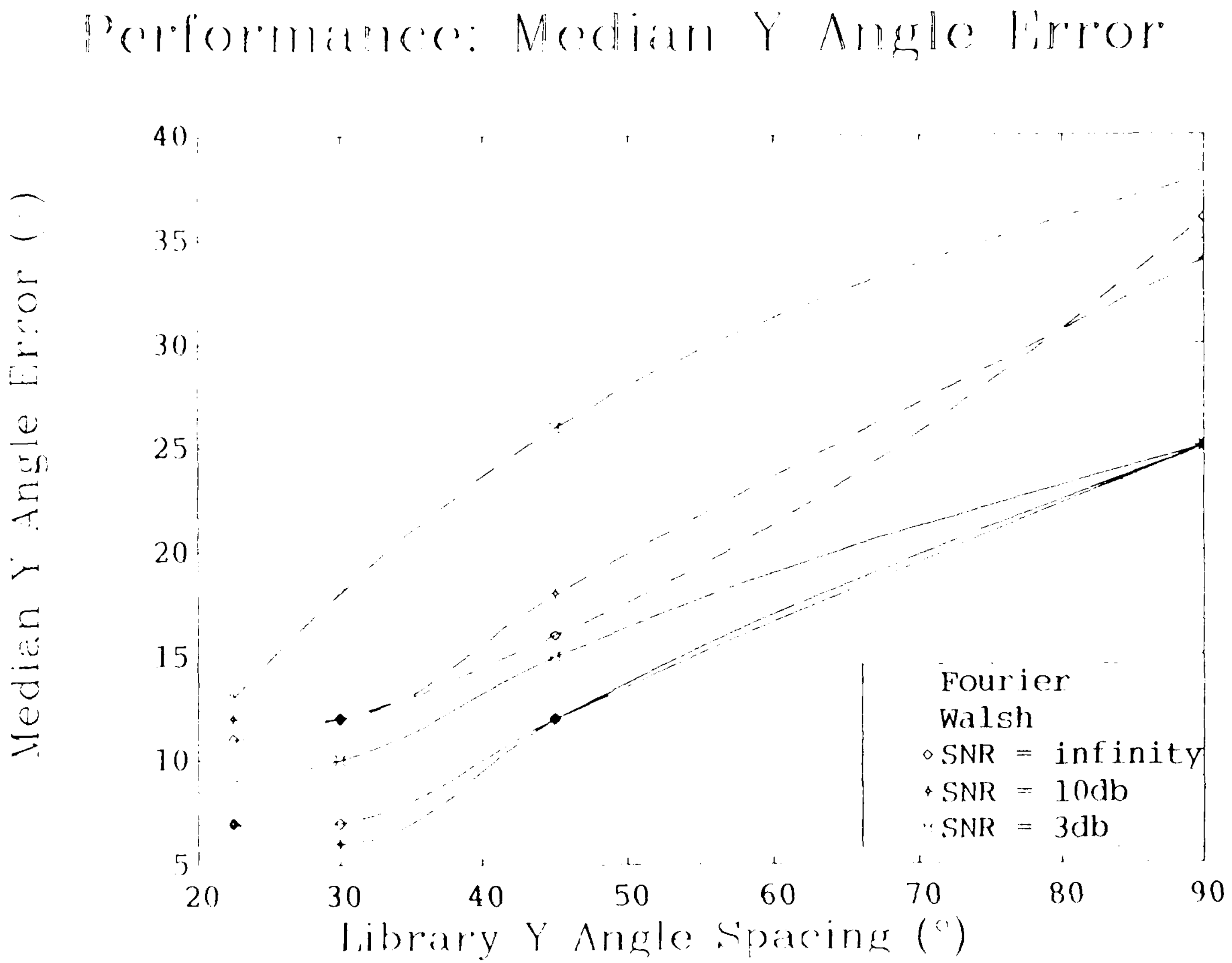


Figure 26: Performance: Median Y Angle Error

these assumptions, the following procedure is proposed to determine rotor orientation.

First, the shape libraries for each helicopter type is enhanced to contain with each view the x and y displacement to the rotor center of rotation. In addition, with each library is included the eight vertices of a rectangular prism which is a bounding surface for the envelope of the rotor blades. These coordinates and displacements are scaled to the reference image size used in the generation of the shape libraries. With each library entry is the scale required for the shape vector normalization. The ratio between the tracked "unknown" contour normalization scale and the stored scale for the assigned library entry orientation provides the scale factor required to scale the displacement vector and bounding surface vertices for the actual scale of the imaged target. The rotation orientation angles θ_x, θ_y, and θ_z are then used to rotate the bounding surface coordinates and then project them onto the image plane. The rotor center of rotation coordinates are rotated in the image plane by θ_z. The outer contour of the scaled rotated, displaced, and projected bounding surface will provide a limited region of interest on which to perform the rest of the operations. Limiting the region of interest will reduce both the total processing time required as well as limit the interfere of clutter in the image.

Once the limits of the imaged bounding surface is obtained, the magnitude and angle of the gradient is computed for all points inside the limits, but not on the silhouette of the helicopter fuselage. A Hough transform [8] is then computed to detect the lines delineating the rotor blades. In computing the Hough transform only pixels inside the bounding region and not inside the contour of the helicopter fuselage (which is readily available) are plotted in Hough space. This is to reduce the inference of linear features interior to and delineating the fuselage contour. The lines detected in Hough space are then followed inside the limits of the bounding region of interest to detect the tips of the rotor blades. The distance from the rotor center of rotation to the detected rotor tips provide the measurements necessary for the final determination of the rotor orientation. If two rotor blade tip locations can be measured, and given the rotor center of revolution, and the scaled length of a rotor blade, the imaged two-dimensional elliptical path of the blade tips can be defined. Given, this information the tilt of the circular disk defining the rotor blade orientation relative to the helicopter fuselage can be determined. The two blade measurements required from the same image can be either the tips of two different blades or the leading and trailing edge of a single blade during the field integration time.

7 Coordinate Transformations

In the following sections the transforms are developed that are necessary to obtain the aircraft orientation (Euler angles) given the observer platform and camera orientation. This discussion begins with the development of the basic transformation matrices, then develops the form of the matrix for an arbitrary observer platform orientation. The basic theory of geometric transformation matrices is taken from Paul [14]. The treatment for a fixed earth centered observer was taken from Brown [1]. This is then extended to the more general condition of an arbitrary (but known) observer orientation as outlined by Gonzalez and Wintz [5].

7.1 Translation Transformation Matrices

Using homogenous coordinates, the translation of a vector, object, or coordinate frame can be performed using matrix multiplication. The translation to the location described by the vector $\mathbf{w} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$, is the displacement a, b, c along the x, y, and z axis, respectively. Multiplication by the following matrix performs this translation:

$$Trans((a,b,c)) = T_{\mathbf{w}} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

7.2 Rotation Transformation Matrices

Rotations of a point, object, or coordinate frame around a coordinate axis can also be performed by multiplying the homogenous coordinates by a matrix. The rotation transformation matrices can be defined for rotation about the x, y, and z-axis. A rotation through a positive angle has the convention that the point is rotated counter-clockwise when observed from the point of view of an observer looking along the axis of rotation towards the origin. The rotation matrices for rotations about the x, y, and z axes are shown below:

$$ROT(x,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$ROT(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$ROT(z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

A sequence of rotations can be performed by simply multiplying the appropriate matrices together. However, the order in when the matrices are multiplied depends on on whether the rotation is performed relative to the base coordinate reference or the transformed coordinates. A rotation about the x-axis followed by a rotation about the y-axis where this latter rotation is about the reference coordinates is described by the matrix product $ROT(y,\theta_y)ROT(x,\theta_x)$. If however, the rotation is about the x-axis followed by a rotation about (the now rotated) body y-axis the matrix product is $ROT(x,\theta_x)ROT(y,\theta_y)$. The first case describes the operations performed in generating the rotated shape libraries for in the image processing frame of reference. The later describes the convention for the Euler angle rotations of an airbourne platform about its body axes.

7.3 Reference Frames

The world coordinate system defined for aircraft orientation is as shown in Figure 27. The positive x-axis points north, the positive y-axis points east, and the positive z-axis points down into the plane of the flat earth. The body axes for the aircraft are embedded in the aircraft such that the origin of the body axes corresponds to the aircraft center of gravity. The orientation of the aircraft body axes are defined as shown in Figure 28. The positive x-axis points out the front of the aircraft nose, the positive y-axis point out to starboard, and the positive z-axis points down through bottom of the aircraft.

The aircraft is in reference orientation when the body axes and the world coordinate axes are parallel. This corresponds to the aircraft in level flight proceeding north.

The coordinate frame for the image processing has been defined differently. The earth centered coordinate axis for the image processing is shown in Figure 29. In this case the positive z-axis points north, the positive x-axis points to the west, and the positive y-axis points out perpendicular to the flat earth.

The body centered coordinates for the image processing are as shown in Figure 30. In this case, the positive z-axis points out the nose, the positive x-axis points to port, and the positive y-axis points out the top of the aircraft.

When the aircraft is in level flight, proceeding north, the earth centered coordinate axes and the body centered coordinate axes are parallel.

In order to describe points, objects, and coordinate frames in the world coordinate system, it will be necessary to transform the image processing coordinates. A rotation of the world coordinates by $\frac{-\pi}{2}$ about the x-axis followed by a rotation about the original z-axis by $\frac{-\pi}{2}$ will transform the world coordinate system into the image processing coordinates. This transformation can be described by the matrix T_E where

$$T_E = ROT(z, \frac{-\pi}{2}) ROT(x, \frac{-\pi}{2}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (9)$$

Premultiplying a point, object, or coordinate frame by the matrix T_E transforms the image processing coordinates to the world coordinates.

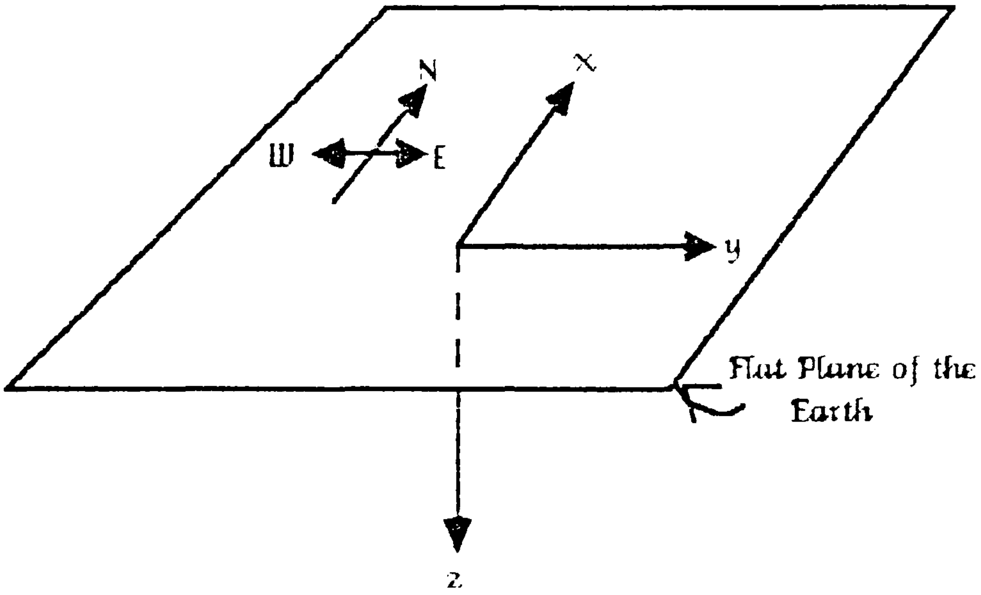


Figure 27: World coordinate reference system

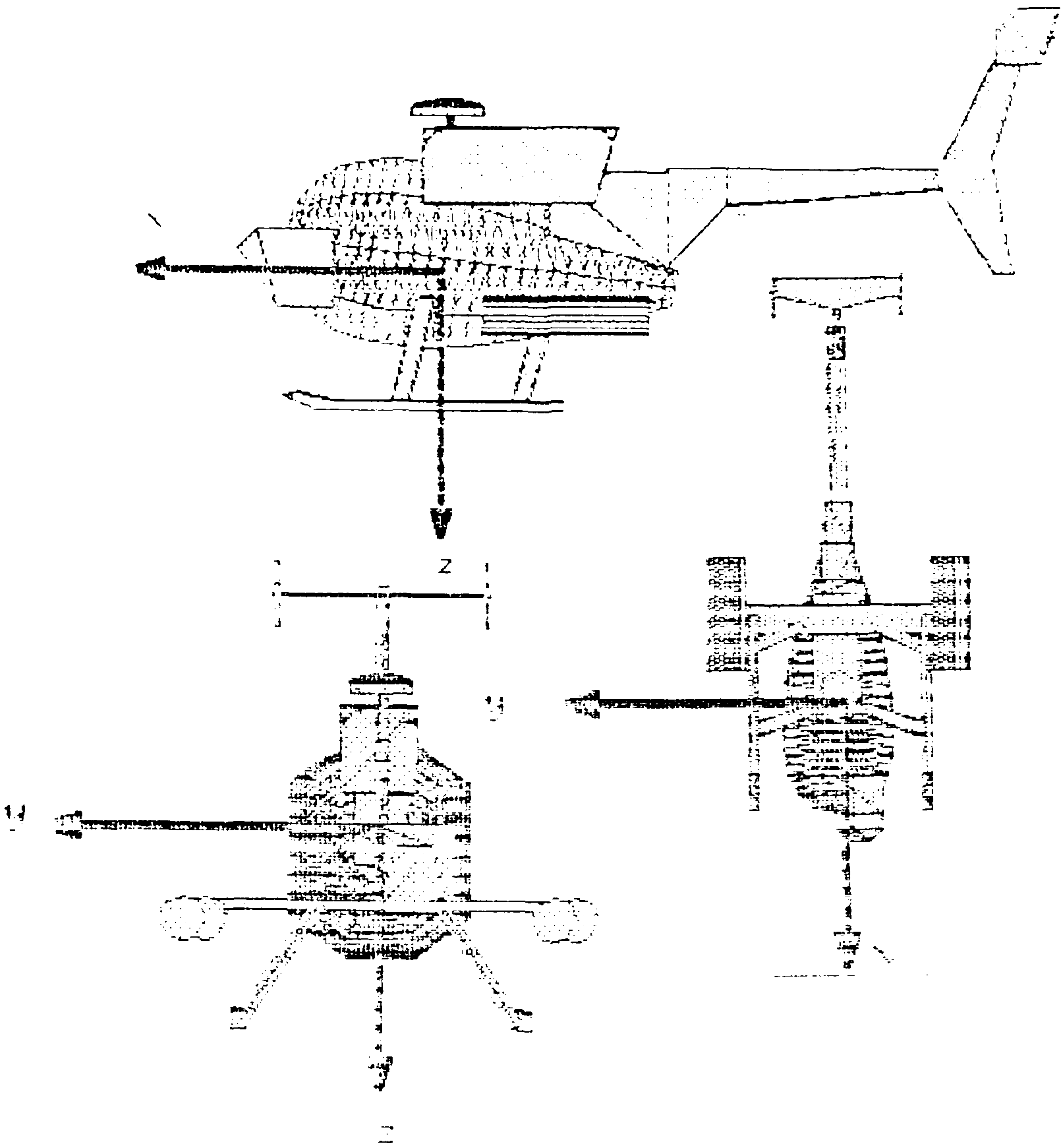


Figure 28: Aircraft body axes coordinate system

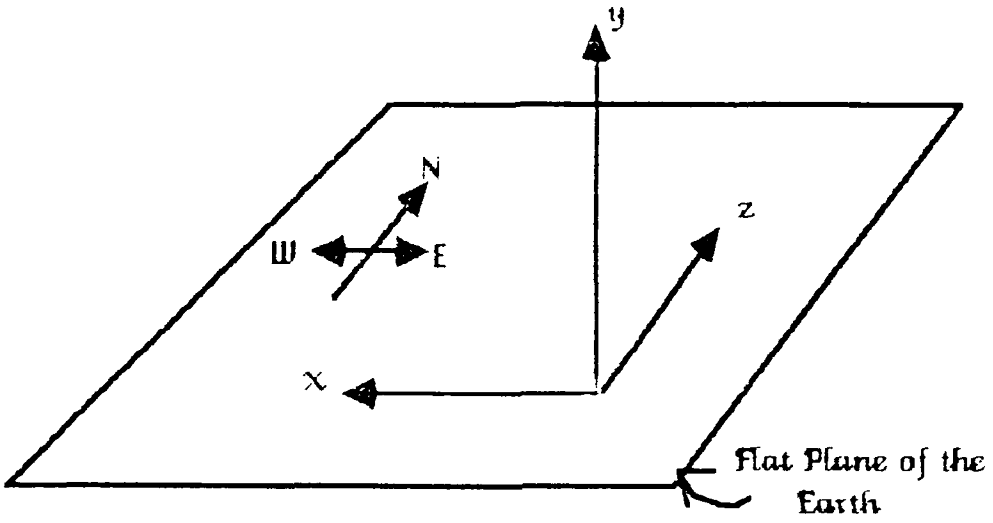


Figure 29: Image processing coordinate reference system

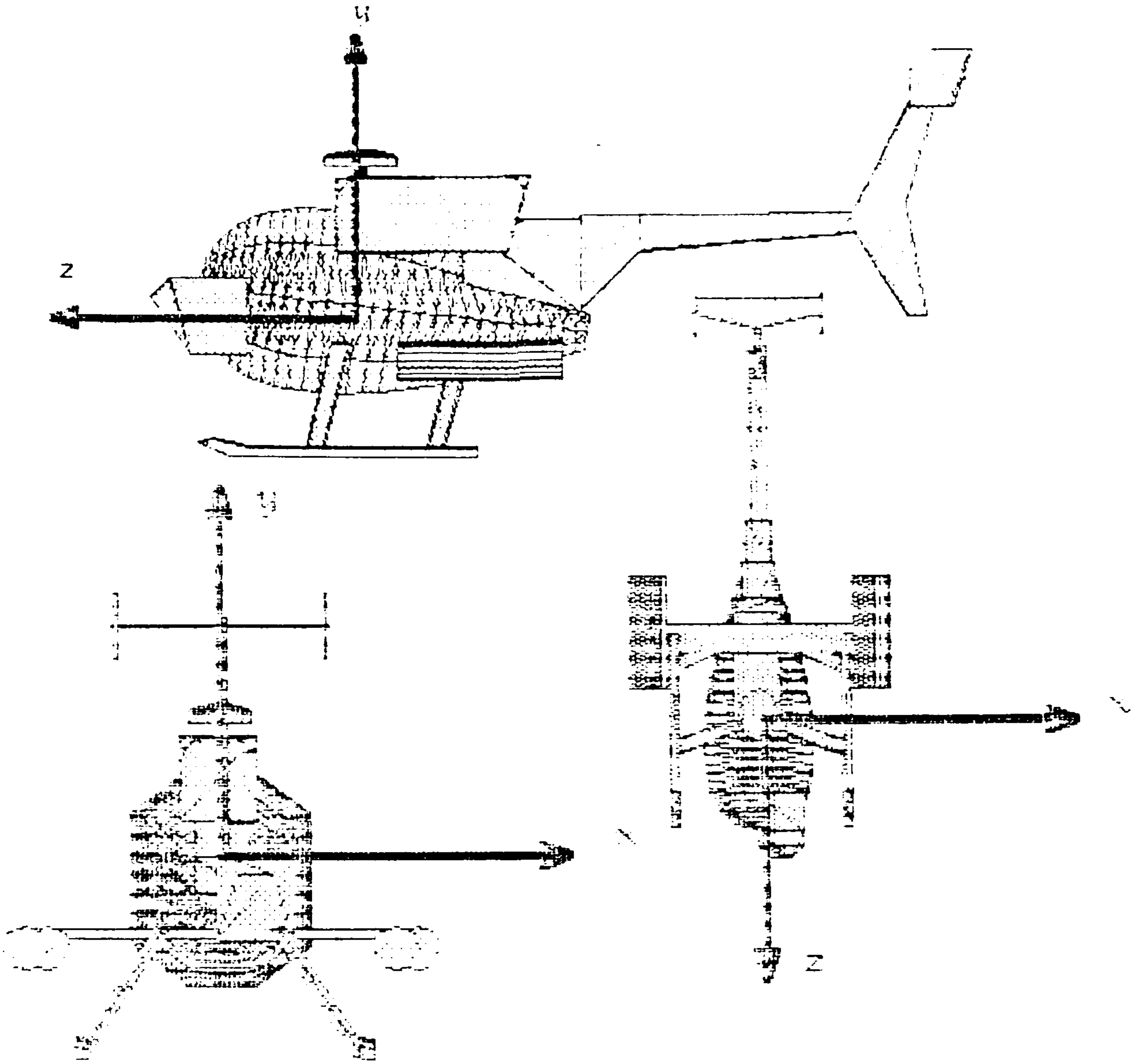


Figure 30: Image processing body axes coordinate system

7.4 Imaging Transformations

Assuming that the camera is centered at the origin of the world coordinate reference system and that the body axes are parallel. The library of views of the aircraft are formed by rotating the object points about the x, y, and z image processing coordinate axes. Let T_x, T_y, and T_z be the rotation transformations about the x, y, and z axes, respectively. That is $T_x = ROT(x, \theta_x)$, $T_y = ROT(y, \theta_y)$, and $T_z = ROT(z, -\theta_z)$. The rotation about the z axis is $-\theta_z$, since the shape matching algorithm measures the rotation from the library reference to the unknown for the z axis rotation.

The $4 \times P$ array of the P homogeneous object coordinates, O, is transformed as follows:

$$O_I^* = T_z T_y T_x O_I \qquad (10)$$

The object coordinates in the image processing reference frame can be transformed to the world coordinates by multiplying by T_E.

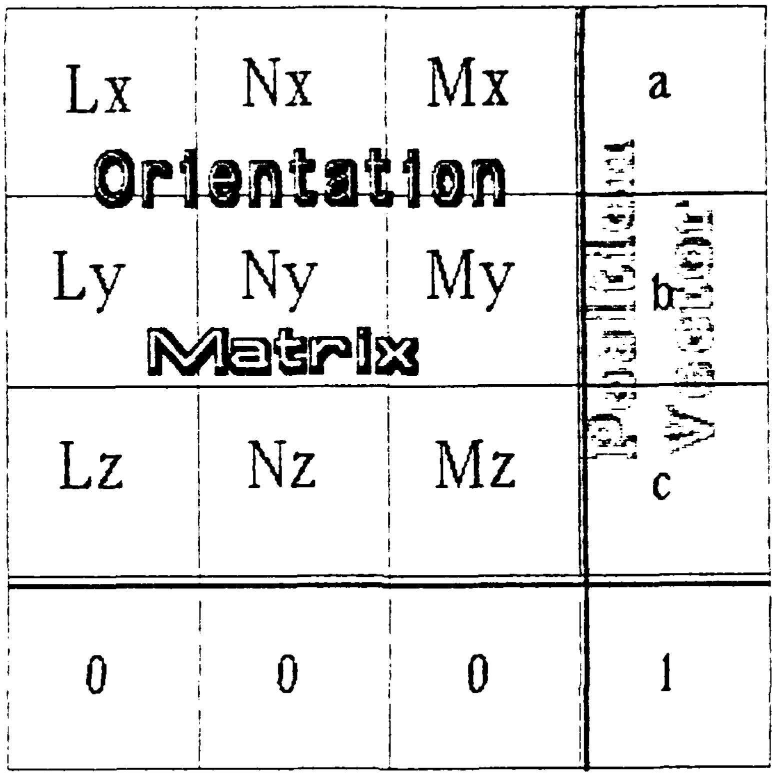
$$\begin{aligned} O_W^* &= T_O O_I \\ T_O &= T_E T_z T_y T_x. \end{aligned} \qquad (11)$$

Finally, the object coordinates are projected on to the image focal plane. The final orientation of the object reference frame is described in the transformation matrix by the elements of the matrix. The composite of all the object transformations can be described by a single matrix having the form

$$T_t = \begin{bmatrix} L_x & M_x & N_x & a \\ L_y & M_y & N_y & b \\ L_z & M_z & N_z & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (12)$$

where (L_x, L_y, L_z) is the transformed unit vector for the object reference x-axis. Similarly, (M_x, M_y, M_z) and (N_x, N_y, N_z) are the transformed y and z axes, respectively. The triplet (a, b, c) is the translation vector. The construction of an homogeneous transformation matrix for object coordinate frame orientation and position is depicted in Figure 31.

The shape matching algorithm provides an estimate of the orientation of the aircraft having undergone the object rotation transformations. The angles θ_x, θ_y, and θ_z can then be used to determine the aircraft body centered orientation (i.e. the Euler angles.)



Lx	Nx	Mx	a
Ly	Ny	My	b
Lz	Nz	Mz	c
0	0	0	1

Figure 31: Construction of a homogeneous transformation

7.5 Euler Angles

The orientation of an aircraft in the world coordinate system is described by its Euler angles. The Euler angles are the rotations of the aircraft *about its body axes*. The sequence of operations used to describe the orientation is first to rotate about the z-axis through the yaw angle, ψ. Then the aircraft is rotated about its *transformed* body y-axis through the pitch angle, θ. Finally, the aircraft is rotated about its transformed body x-axis through the roll angle, ϕ. Since these rotations are performed on the transformed coordinates, the composite rotation transformation is the matrix $T = T_\psi T_\theta T_\phi$ where $T_\psi = ROT(z, \psi)$,$T_\theta = ROT(y, \theta)$, and $T_\phi = ROT(x, \phi)$.

The aircraft being tracked by the image processing system has first been rotated through some unknown Euler angles and then the body centered axes are translated to the position $\mathbf{w}_t$. Therefore, the tracked object points relative to the world coordinate system, $O_{\mathbf{w}}$, are transformed to

$$O^*_{\mathbf{w}} = T_{\mathbf{w}_t} T_{\psi_t} T_{\theta_t} T_{\phi_t} O_{\mathbf{w}}. \qquad (13)$$

The Euler angles for the tracked aircraft can be determined by equating the elements of the orientation sub-matrix for the transformation on the tracked object, $T_{\mathbf{w}_t} T_{\psi_t} T_{\theta_t} T_{\phi_t}$ and the orientation sub-matrix for the observed transformation, $T_E T_z T_y T_x$. This assumes the camera is at the origin of the world reference system with it coordinate axes parallel with the world coordinate axes.

7.6 Observer Coordinate Transformations

In most real circumstances the camera is not located at the world coordinate system origin. Neither is the camera coordinate axes and the world coordinate axes parallel. The camera is positioned onboard another airbourne platform. As this platform moves the camera orientation changes. It is therefore necessary to compensate for the fact the the camera is not in the reference position.

First, the operations performed to place the camera in its actual position and orientation will be discussed. Then the transformations necessary to compensate for this change observed orientation of the tracked object will be presented.

The imaging system is mounted on a gimbal assembly at some position on the outside of the aircraft. Let $\mathbf{G} = (g_x, g_y, g_z)$ represent the vector displacement of the gimbal assembly pivot point from the aircraft center of gravity. Let the displacement of the imaging focal plane center from the gimbal pivot point be described by the vector $\mathbf{f} = (f_x, f_y, f_z)$. This translation transformation can each be performed using the following matrices:

$$T_{\mathrm{G}} = \begin{bmatrix} 1 & 0 & 0 & g_x \\ 0 & 1 & 0 & g_y \\ 0 & 0 & 1 & g_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

and

$$T_{\mathbf{f}} = \begin{bmatrix} 1 & 0 & 0 & f_x \\ 0 & 1 & 0 & f_y \\ 0 & 0 & 1 & f_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

The gimbal assembly allows the camera to be panned in azimuth. This corresponds to a rotation about the z-axis through an angle θ_{az}. The camera can be rotated in elevation corresponding to a rotation of angle θ_{el} about the y-axis. Let $T_{\theta_{az}}$ and $T_{\theta_{el}}$ represent the rotation transformations for the rotation of the camera in azimuth and elevation, respectively.

The aircraft platform on which the camera is mounted under goes rotation of its body axes as described by the Euler angle transformation matrices $T_{\psi_c} T_{\theta_c} T_{\phi_c}$. Lastly, the position of the platform relative to the world coordinate system is described as a translation transformation $T_{\mathbf{w_c}}$.

The camera coordinate frame therefore undergoes the following transformation:

$$O_{\mathbf{w}}^{*c} = T_c O_{\mathbf{w}}^c,$$

where

$$T_c = T_{\mathbf{w_c}} T_{\psi_c} T_{\theta_c} T_{\phi_c} T_G T_{\theta_{el}} T_{\theta_{az}} T_{\mathbf{f}}. \tag{16}$$

The transformations on the camera coordinate frame does not move any of the points in the world coordinate system including the points of the object being tracked.

The discussion for determing the tracked object Euler angles from the observed rotation angles was based on the assumption that the camera was positioned at the origin of the world coordinate system with the axes parallel with the world coordinate axes. The camera can be placed in standard position by performing the inverse transformation on the tracked object coordinate frame. Therefore, the tracked object transformation matrix is

$$T_t = T_c^{-1} T_{\mathbf{w}_t} T_{\psi_t} T_{\theta_t} T_{\phi_t}. \tag{17}$$

To determine the Euler angles given the observed imaging system rotation angles, the camera elevation and azimuth angles, the observation aircraft platform position & orientation, and the displacements to the gimbal pivot point & gimbal pivot to image plane. it is necessary to equate the elements of the orientation sub-matrix, T_t and that of the observed orientation sub-matrix, T_O.

$$T_c^{-1} T_{\mathbf{w}_t} T_{\psi_t} T_{\theta_t} T_{\phi_t} = T_E T_z T_y T_x.$$

or

$$T_{\psi_t} T_{\theta_t} T_{\phi_t} = T_{\mathbf{w}_t}^{-1} T_c T_E T_z T_y T_x. \tag{18}$$

The elements of the orientation sub-matrix of the left-hand side of the above equation are given in Appendix A. The expressions were computed using the symbolic mathematics processing program REDUCE [7].

Letting T_t represent the matrix on the right-hand side of the equation, we can now solve for the tracked objects Euler angles.

So, we have

$$T_{\psi_t} T_{\theta_t} T_{\phi_t} = T_t$$

or

$$T_{\theta_t} T_{\phi_t} = T_{\psi_t}^{-1} T_t. \tag{19}$$

Performing the matrix multiplications, we then have

$$\begin{bmatrix} \cos\theta_t & \sin\phi_t \sin\theta_t & \cos\phi_t \sin\theta_t \\ 0 & \cos\phi_t & -\sin\phi_t \\ -\sin\theta_t & \cos\theta_t \sin\phi_t & \cos\phi_t \cos\theta_t \end{bmatrix}$$

$$= \begin{bmatrix} L_x \cos\psi_t - L_y \sin\psi_t & M_x \cos\psi_t - M_y \sin\psi_t & N_x \cos\psi_t - N_y \sin\psi_t \\ L_y \cos\psi_t + L_x \sin\psi_t & M_y \cos\psi_t + M_x \sin\psi_t & N_y \cos\psi_t + N_x \sin\psi_t \\ L_z & M_z & N_z \end{bmatrix} \tag{20}$$

Now, equate corresponding elements in the two matrices. First, we have

$$L_y \cos \psi_t + L_x \sin \psi_t = 0.$$

So,

$$\tan \psi_t^{1,2} = \frac{\mp L_y}{\pm L_x}$$

Therefore, we have two possible solutions for ψ_t

$$\psi_t^{1,2} = \operatorname{atan2}(\mp L_y, \pm L_x). \tag{21}$$

where ψ_t^1 and ψ_t^2 differ by π. Having,

$$\sin \theta_t = -L_z \text{ and } \cos \theta_t = L_x \cos \psi_t - L_y \sin \psi_t$$

gives

$$\theta_t = \operatorname{atan2}(\mp L_z, \pm(L_x \cos \psi_t^{1,2} - L_y \sin \psi_t^{1,2}). \tag{22}$$

Lastly,

$$\sin \phi_t = -N_y \cos \psi_t - N_x \sin \psi_t \text{ and } \cos \phi_t = M_y.$$

Therefore,

$$\phi_t = \operatorname{atan2}(\mp N_y \cos \psi_t^{1,2} \mp N_x \sin \psi_t^{1,2}, \pm M_y \cos \psi_t^{1,2} \pm M_x \sin \psi_t^{1,2}) \tag{23}$$

Since θ_t and ϕ_t are dependant upon ψ_t, choose ψ_t, θ_t and ϕ_t such that

$$-\pi \le \psi_t \le \pi \text{ and } -\frac{\pi}{2} \le \theta_t \le \frac{\pi}{2} \text{ and } -\pi \le \phi_t \le \pi.$$

When the aircraft is in vertical dive or climb, then $L_y = L_x = 0$ and psi_t is undefined. In this case, $\psi_t = 0$ by convention.

8 Conclusions

In this study the results are presented for the comparing the use of two global shape recognition algorithms applied to the problem of helicopter type classification and orientation determination. Under the conditions of varying library density and simulated imaging noise, it is shown that the Fourier descriptor method is superior in performance measured as classification accuracy and mean (or median) angle error.

In addition, a scheme is proposed describing outlining the library, segmentation, and processes for the determination of rotor tip path. The relative orientation of the blade path disk can be determined from the elliptical shape of the projected rotor tip path.

Finally, the transformations for computing the tracked object Euler (orientation) angles are derived.

9 References

1. Robin E. Brown, "Transformation of Observed Aircraft Orientation to Euler Angles," U.S. Army Summer Assoc. Program for High School Science and Mathematics Faculty, Aug. 22, 1986.

2. Charles R. Giardina and Frank P. Kuhl, "Accuracy of Curve Approximation by Harmonically Related Vectors with Elliptical Loci," *Computer Graphics and Image Processing 6*, pp. 277-285, 1977.

3. Charles R. Giardina, Frank P. Kuhl, T.A. Grogan, and O. Robert Mitchell, "A Spatial Domain Walsh Feature Set," *Trans. of the 28th Conference of Army Mathematicians*, ARO Report 83-1, pp. 397-408, February, 1983.

4. Marcus Elgin Glenn, "Fourier Descriptors of Two and Three Dimensional Objects," *M.S.E.E. Thesis*, Purdue University, May, 1982.

5. Rafael C. Gonzalez and Paul Wintz, *Digital Image Processing*, Second Edition, Addison-Wesley, 1987.

6. Timothy A. Grogan, "Shape Recognition and Description: A Comparative Study," *Ph.D.E.E. Thesis*, Purdue University, August, 1983.

7. Anthony C. Hearn, "REDUCE User's Manual, Version 3.0", *Rand Publication* CP78(4/83), 1983.

8. P.V.C. Hough, "Methods and Means for Recognizing Complex Patterns," U.S. Patent 3,069,654, 1962.

9. *Jane's All the Worlds Aircraft*,pp. 335, 337, 462, 463, 510, 511, (1985).

10. Frank P. Kuhl and Charles R. Giardina, "Elliptic Fourier Features of a Closed Contour," *Computer Graphics and Image Processing*, Vol. 18, pp. 236-258, 1982.

11. Mac3D, Version 2.0, Challenger Software Corporation, 1985.

12. O. Robert Mitchell, Frank P. Kuhl, Timothy A. Grogan, and Didier Charpentier, "A Shape Extraction and Recognition System," *Southcon/82*, March 23-25, Orlando, Florida, pp.(4-1) 1-4, 1982.

13. O. Robert Mitchell, Anthony P. Reeves, and Timothy A. Grogan, "Algorithms and Architectures for Global Shape Analysis in Time-Varying Imagery," *SPIE Proceedings 360, Robotics and Industrial Applications*, San Diego, Aug. 24-27, 1982.

14. Richard P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, 1981.

15. E. Persoon and K.S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, Vol. SMC-7, pp. 170-179, March, 1977.

16. Timothy P. Wallace and P.A. Wintz, "An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors," *Computer Graphics and Image Processing*, Vol. 13, pp. 99-126, 1980.

17. Timothy P. Wallace and Owen R. Mitchell, "Analysis of three-dimensional movement using Fourier descriptors," *IEEE Trans. Pattern Anal. and Mach. Intell.*, Vol. PAMI-2, no. 6, pp. 583-588, November, 1980.

APPENDIX A

Raw Data for Classification and Orientation Accuracy

The following pages contain the raw data for the experiments determining the classification and orientation accuracy for the helicopter fuselages. The information is organized into three groups results: those for the Fourier descriptors, the Walsh points, and Fourier descriptors using multiple normalizations. Then for each group, the results are tabulated under the variations in the library density and the "unknown" image signal-to-noise ratio.

The libraries are numbered 1, 2, and 3, corresponding to the Bell Model 205 UH-1, McDonnelll Douglas 500MD, and Sikorsky UH60A Black Hawk, respectively.

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   63     0     0
     2    0    62     1
     3    0     0    63

 correct classification   188/  189 which is   99.47%


                    188 Hits                  1 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               25.0      6.9            165.0      2.0
Median              8.0      7.0            165.0      2.0
Std. Devn.         40.5      4.2              0.0      0.0

  Library Size 11x9   SNR inf   Features Fourier_Coeffs
```

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   63     0     0
     2    0    62     1
     3    0     0    63

 correct classification   188/  189 which is   99.47%


                    188 Hits                  1 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               27.2      7.0            165.0      2.0
Median              9.0      7.0            165.0      2.0
Std. Devn.         43.6      4.3              0.0      0.0

  Library Size 11x9   SNR 10   Features Fourier Coeffs
```

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   60     0     3
     2    3    57     3
     3    2     0    61

 correct classification   178/  189 which is   94.18%


                    178 Hits                 11 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               33.9     12.5             81.3     38.4
Median             14.0      9.0             92.0     28.0
Std. Devn.         41.3     14.1             62.8     41.9

  Library Size 11x9   SNR 3   Features Fourier Coeffs
```

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   62     0     1
     2    1    61     1
     3    1     0    62

 correct classification   185/  189 which is   97.88%


                    185 Hits                  4 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               26.6      8.5            119.8     12.8
Median             10.0      7.0            123.0      2.0
Std. Devn.         41.3      6.3             47.3     19.2

  Library Size_9x7   SNR_inf   Features_Fourier_Coeffs
```

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   62     0     1
     2    2    60     1
     3    1     0    62

 correct classification   184/  189 which is   97.35%


                    184 Hits                  5 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               26.3      8.6            111.4     24.8
Median             10.0      6.0            123.0      2.0
Std. Devn.         39.5      6.5             45.5     29.6

  Library Size 9x7   SNR 10   Features Fourier Coeffs
```

```
 Confusion Matrix

 each row gives the classification for one class
          1     2     3
     1   61     0     2
     2    4    54     5
     3    4     0    59

 correct classification   174/  189 which is   92.06%


                    174 Hits                 15 Misses
Error            X angle  Y angle          X angle  Y angle
Mean               38.4     13.4             60.1     28.1
Median             18.0     10.0             51.0     11.0
Std. Devn.         46.2     12.0             51.4     37.5

  Library Size 9x7   SNR 3   Features Fourier Coeffs
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   61     0     2
    2    3    59     1
    3    2     0    61

correct classification   181/  189 which is   95.77%

                   181 Hits               8 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              36.8     13.9         82.3     19.6
Median            17.0     12.0         70.0      5.0
Std. Devn.        45.1     10.6         58.2     22.9

  Library Size 7x5   SNR inf   Features Fourier Coeffs
```

```
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   61     0     2
    2    3    59     1
    3    2     0    61

correct classification   181/  189 which is   95.77%

                   181 Hits               8 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              38.0     14.0         96.8     25.3
Median            17.0     12.0         70.0     16.0
Std. Devn.        46.6     10.7         61.2     23.4

  Library Size 7x5   SNR 10   Features Fourier Coeffs
```

```
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   58     0     5
    2    7    51     5
    3    3     0    60

correct classification   169/  189 which is   89.42%

                   169 Hits               20 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              45.3     17.2         66.1     39.2
Median            22.0     15.0         53.0     26.0
Std. Devn         43.3     13.4         51.0     38.0

  Library Size 7x5   SNR 3   Features Fourier Coeffs
```

```
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   51    11     1
    2    6    54     3
    3    3     0    60

correct classification   165/  189 which is   87.30%

                   165 Hits               24 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              73.1     25.2         81.3     44.0
Median            67.0     25.0         67.0     48.0
Std. Devn         54.9     20.5         48.9     25.1

  Library Size 5x3   SNR inf   Features Fourier Coeffs
```

```
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   48    13     2
    2    6    54     3
    3    4     0    59

correct classification   161/  189 which is   85.19%

                   161 Hits               28 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              74.1     25.2         83.3     46.3
Median            68.0     25.0         71.0     48.0
Std. Devn         55.1     21.6         46.9     27.9

  Library Size 5x3   SNR 10   Features Fourier Coeffs
```

```
 Confusion Matrix

each row gives the classification for one class
        1     2     3
    1   52     8     3
    2    7    48     8
    3    3     1    59

correct classification   159/  189 which is   84.13%

                   159 Hits               30 Misses
Error            X angle  Y angle       X angle  Y angle
Mean              63.6     26.1         82.8     42.3
Median            51.0     25.0         78.0     36.0
Std. Devn         51.1     22.8         48.2     30.4

  Library Size 5x3   SNR 3   Features Fourier Coeffs
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    49     2    12
    2    10    51     2
    3     2     0    61

correct classification   161/  189 which is   85.19%


                    161 Hits                 28 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             58.4      21.0           54.2      56.1
Median           36.0      11.0           31.0      48.0
Std. Devn        55.1      25.7           49.5      33.6

  Library Size 11x9   SNR_inf   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    48     4    11
    2     9    51     3
    3     4     0    59

correct classification   158/  189 which is   83.60%


                    158 Hits                 31 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             56.3      21.1           56.5      59.9
Median           31.0      12.0           37.0      53.0
Std. Devn.       54.6      24.7           47.8      34.1

  Library_Size 11x9   SNR_10   Features Walsh_Points
```

```
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    37    11    15
    2    14    45     4
    3     7     0    56

correct classification   138/  189 which is   73.02%


                    138 Hits                 51 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             62.8      24.2           61.5      57.0
Median           44.0      13.0           54.0      53.0
Std. Devn        52.3      27.0           43.5      31.2

  Library Size 11x9   SNR 3   Features Walsh_Points
```

```
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    50     5     8
    2     9    51     3
    3     2     0    61

correct classification   162/  189 which is   85.71%


                    162 Hits                 27 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             57.6      19.1           64.0      54.6
Median           31.0      12.0           54.0      50.0
Std. Devn.       57.0      22.9           51.8      28.1

  Library_Size 9x7   SNR_inf   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    49     5     9
    2     9    52     2
    3     3     0    60

correct classification   161/  189 which is   85.19%


                    161 Hits                 28 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             59.0      19.1           50.1      50.1
Median           35.0      12.0           30.0      46.0
Std. Devn.       56.2      22.3           42.7      25.3

  Library Size 9x7   SNR_10   Features_Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
          1     2     3
    1    39    11    13
    2    14    43     6
    3     9     4    50

correct classification   132/  189 which is   69.84%


                    132 Hits                 57 Misses
Error           X angle   Y angle        X angle   Y angle
Mean             66.4      26.1           66.0      47.0
Median           51.0      18.0           48.0      44.0
Std. Devn.       55.0      26.8           50.8      30.9

  Library Size 9x7   SNR 3   Features Walsh Points
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   47     5   11
     2   13    43    7
     3    2     2   59

correct classification   149/  189 which is   78.84%

                    149 Hits               40 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               65.2     28.2          57.5     60.0
Median             44.0     16.0          43.0     65.0
Std. Devn          46.9     28.3          48.8     33.5

  Library Size 7x5   SNR inf   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   47     7    9
     2   15    41    7
     3    3     2   58

correct classification   146/  189 which is   77.25%

                    146 Hits               43 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               58.1     31.5          63.8     56.1
Median             49.0     18.0          49.0     62.0
Std. Devn          41.9     31.7          47.6     34.5

  Library Size 7x5   SNR 10   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   38    11   14
     2   20    36    7
     3    8     4   51

correct classification   125/  189 which is   66.14%

                    125 Hits               64 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               60.3     33.2          62.8     53.1
Median             47.0     26.0          49.0     41.0
Std. Devn          40.3     32.5          49.0     46.2

  Library Size 7x5   SNR 3   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   48     8    7
     2   25    31    7
     3    3     5   55

correct classification   134/  189 which is   70.90%

                    134 Hits               55 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               70.5     50.4          69.7     77.8
Median             59.0     36.0          52.0     74.0
Std. Devn.         51.0     43.3          54.5     48.4

  Library Size 5x3   SNR inf   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   47     8    8
     2   22    33    8
     3    9     4   50

correct classification   130/  189 which is   68.78%

                    130 Hits               59 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               70.6     53.3          77.1     72.3
Median             62.9     34.0          67.0     71.0
Std. Devn          49.8     48.2          53.3     43.8

  Library Size 5x3   SNR 10   Features Walsh Points
```

```
 Confusion Matrix

each row gives the classification for one class
         1     2     3
     1   36    16   11
     2   23    33    7
     3   10     4   49

correct classification   118/  189 which is   62.43%

                    118 Hits               71 Misses
Error             X angle  Y angle       X angle  Y angle
Mean               67.3     55.3          78.9     67.8
Median             59.9     38.0          67.0     64.0
Std. Devn          50.4     46.4          53.5     46.2

  Library Size 5x3   SNR 3   Features Walsh Points
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    63     0     0
     2     0    63     0
     3     0     0    63

correct classification   189/  189 which is  100.00%

                     189 Hits                  0 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               22.3      7.0            0.0       0.0
Median              8.0      7.0            0.0       0.0
Std. Devn.         36.3      4.1            0.0       0.0

Library Size 11x9 SNR inf Features Fourier (Mult. Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    63     0     0
     2     0    63     0
     3     1     0    62

correct classification   188/  189 which is   99.47%

                     188 Hits                  1 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               25.5      6.9          153.0       2.0
Median              8.0      7.0          153.0       2.0
Std. Devn.         41.0      4.2            0.0       0.0

Library Size 11x9 SNR 10 Features Fourier (Mult  Normlzn )
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    63     0     0
     2     2    60     1
     3     1     0    62

correct classification   185/  189 which is   97.88%

                     185 Hits                  4 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               34.4     11.7           46.3      13.5
Median             14.0      9.0           14.0       2.0
Std. Devn.         43.2     11.9           61.8      16.6

Library Size 11x9 SNR 3 Features Fourier (Mult. Normlzn )
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    63     0     0
     2     1    61     1
     3     2     0    61

correct classification   185/  189 which is   97.88%

                     185 Hits                  4 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               25.7      8.4           53.3      13.0
Median             10.0      6.0           15.0       2.0
Std. Devn.         39.9      6.2           56.8      19.1

Library_Size 9x7 SNR inf Features_Fourier_(Mult._Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    63     0     0
     2     2    60     1
     3     2     0    61

correct classification   184/  189 which is   97.35%

                     184 Hits                  5 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               26.1      8.2           58.2      25.0
Median             10.0      6.0           42.0       2.0
Std. Devn.         39.6      5.8           51.7      29.4

Library Size 9x7 SNR 10 Features Fourier (Mult. Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    62     0     1
     2     4    57     2
     3     1     0    62

correct classification   181/  189 which is   95.77%

                     181 Hits                  8 Misses
Error             X angle  Y angle         X angle  Y angle
Mean               37.1     12.8           60.8      31.3
Median             17.0     10.0           48.0      11.0
Std. Devn.         45.9     11.6           45.5      40.7

Library Size 9x7 SNR 3 Features Fourier (Mult. Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    62     0     1
     2     3    59     1
     3     1     0    62

correct classification   183/  189 which is   96.83%


                   183 Hits                 6 Misses
Error            X angle  Y angle        X angle  Y angle
Mean              36.1     13.7           93.5     22.7
Median            17.0     12.0           70.0      2.0
Std. Devn.        44.5     10.4           56.7     25.5

Library Size 7x5 SNR inf Features Fourier (Mult. Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    62     0     1
     2     3    59     1
     3     1     0    62

correct classification   183/  189 which is   96.83%


                   183 Hits                 6 Misses
Error            X angle  Y angle        X angle  Y angle
Mean              37.3     13.7           85.8     22.7
Median            17.0     12.0           57.0      2.0
Std. Devn         46.0     10.4           60.6     25.5

Library Size 7x5 SNR 10 Features Fourier (Mult. Normlzn )
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    58     0     5
     2     7    52     4
     3     1     0    62

correct classification   172/  189 which is   91.01%


                   172 Hits                 17 Misses
Error            X angle  Y angle        X angle  Y angle
Mean              45.9     16.8           52.9     33.3
Median            22.0     14.0           40.0     25.0
Std. Devn         50.4     13.2           46.9     35.2

Library Size 7x5 SNR 3 Features Fourier (Mult. Normlzn )
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    50    12     1
     2     7    53     3
     3     1     0    62

correct classification   165/  189 which is   87.30%


                   165 Hits                 24 Misses
Error            X angle  Y angle        X angle  Y angle
Mean              72.0     24.6           80.9     45.3
Median            67.0     25.0           67.0     48.0
Std. Devn.        54.6     20.1           48.4     23.6

Library Size 5x3 SNR inf Features Fourier (Mult._Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    48    14     1
     2     7    54     2
     3     2     0    61

correct classification   163/  189 which is   86.24%


                   163 Hits                 26 Misses
Error            X angle  Y angle        X angle  Y angle
Mean              73.8     24.8           78.8     43.7
Median            68.0     25.0           67.0     47.0
Std. Devn.        55.0     21.4           45.6     23.1

Library Size 5x3 SNR 10 Features Fourier (Mult. Normlzn.)
```

```
 Confusion Matrix

each row gives the classification for one class
           1     2     3
     1    54     8     1
     2    10    50     3
     3     3     1    59

correct classification   163/  189 which is   86.24%


                   163 Hits                 26 Misses
Error            X angle  Y angle        X arqle  Y angle
Mean              67.8     25.2           73.2     36.6
Median            55.0     20.0           72.0     36.0
Std. Devn         53.2     21.8           46.1     19.9

Library Size 5x3 SNR 3 Features Fourier (Mult. Normlzn.)
```

APPENDIX B

Object Orientation Matrix

Below are the expressions for the elements of the object orientation matrix as computed using the REDUCE [7] program. The orientation matrix is the upper 3x3 submatrix of the matrix formed by the transformation

$$T_{\mathbf{w_t}} T_{\mathbf{w_c}} T_{\psi_c} T_{\theta_c} T_{\psi_c} T_G T_{\theta_e l} T_{\theta_a z} T_{\mathbf{f}} T_E T_x T_y T_z$$

The correspondence between the parameters of the above transformation and the symbols used in the listing are shown below:

θ_x	=	thetax
θ_y	=	thetay
θ_z	=	thetaz
ψ_c	=	psic
θ_c	=	thetac
ϕ_c	=	phic
$\theta_e l$	=	thetael
$\theta_a z$	=	thetaaz

```
Lx      =      cos(phic)*cos(psic)*cos(thetaaz)*sin(thetac)*sin(thetael)*

               sin(thetay) + cos(phic)*cos(psic)*cos(thetael)*cos(thetay)

               *sin(thetac)*sin(thetaz) + cos(phic)*cos(psic)*cos(thetay)

               *cos(thetaz)*sin(thetaaz)*sin(thetac)*sin(thetael) - cos(

               phic)*cos(thetaaz)*cos(thetay)*cos(thetaz)*sin(psic) + cos

               (phic)*sin(psic)*sin(thetaaz)*sin(thetay) - cos(psic)*cos(

               thetaaz)*cos(thetac)*cos(thetael)*sin(thetay) - cos(psic)*

               cos(thetaaz)*cos(thetay)*cos(thetaz)*sin(phic)*sin(thetac)

                - cos(psic)*cos(thetac)*cos(thetael)*cos(thetay)*cos(

               thetaz)*sin(thetaaz) + cos(psic)*cos(thetac)*cos(thetay)*

               sin(thetael)*sin(thetaz) + cos(psic)*sin(phic)*sin(thetaaz
                  )*sin(thetac)*sin(thetay) - cos(thetaaz)*sin(phic)*sin(

                  psic)*sin(thetael)*sin(thetay) - cos(thetael)*cos(thetay)*

                  sin(phic)*sin(psic)*sin(thetaz) - cos(thetay)*cos(thetaz)*

                  sin(phic)*sin(psic)*sin(thetaaz)*sin(thetael)

Mx     =          - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetay)*sin(thetac

                  )*sin(thetael)*sin(thetax) - cos(phic)*cos(psic)*cos(

                  thetael)*cos(thetax)*cos(thetaz)*sin(thetac) + cos(phic)*

                  cos(psic)*cos(thetael)*sin(thetac)*sin(thetax)*sin(thetay)

                  *sin(thetaz) + cos(phic)*cos(psic)*cos(thetax)*sin(thetaaz

                  )*sin(thetac)*sin(thetael)*sin(thetaz) + cos(phic)*cos(

                  psic)*cos(thetaz)*sin(thetaaz)*sin(thetac)*sin(thetael)*

                  sin(thetax)*sin(thetay) - cos(phic)*cos(thetaaz)*cos(

                  thetax)*sin(psic)*sin(thetaz) - cos(phic)*cos(thetaaz)*cos

                  (thetaz)*sin(psic)*sin(thetax)*sin(thetay) - cos(phic)*cos

                  (thetay)*sin(psic)*sin(thetaaz)*sin(thetax) + cos(psic)*

                  cos(thetaaz)*cos(thetac)*cos(thetael)*cos(thetay)*sin(

                  thetax) - cos(psic)*cos(thetaaz)*cos(thetax)*sin(phic)*sin

                  (thetac)*sin(thetaz) - cos(psic)*cos(thetaaz)*cos(thetaz)*

                  sin(phic)*sin(thetac)*sin(thetax)*sin(thetay) - cos(psic)*

                  cos(thetac)*cos(thetael)*cos(thetax)*sin(thetaaz)*sin(

                  thetaz) - cos(psic)*cos(thetac)*cos(thetael)*cos(thetaz)*

                  sin(thetaaz)*sin(thetax)*sin(thetay) - cos(psic)*cos(
            thetac)*cos(thetax)*cos(thetaz)*sin(thetael) + cos(psic)*

            cos(thetac)*sin(thetael)*sin(thetax)*sin(thetay)*sin(

            thetaz) - cos(psic)*cos(thetay)*sin(phic)*sin(thetaaz)*sin

            (thetac)*sin(thetax) + cos(thetaaz)*cos(thetay)*sin(phic)*

            sin(psic)*sin(thetael)*sin(thetax) + cos(thetael)*cos(

            thetax)*cos(thetaz)*sin(phic)*sin(psic) - cos(thetael)*sin

            (phic)*sin(psic)*sin(thetax)*sin(thetay)*sin(thetaz) - cos

            (thetax)*sin(phic)*sin(psic)*sin(thetaaz)*sin(thetael)*sin

            (thetaz) - cos(thetaz)*sin(phic)*sin(psic)*sin(thetaaz)*

            sin(thetael)*sin(thetax)*sin(thetay)

Mx   =      - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetax)*cos(thetay

            )*sin(thetac)*sin(thetael) + cos(phic)*cos(psic)*cos(

            thetael)*cos(thetax)*sin(thetac)*sin(thetay)*sin(thetaz)

             + cos(phic)*cos(psic)*cos(thetael)*cos(thetaz)*sin(thetac

            )*sin(thetax) + cos(phic)*cos(psic)*cos(thetax)*cos(thetaz

            )*sin(thetaaz)*sin(thetac)*sin(thetael)*sin(thetay) - cos(

            phic)*cos(psic)*sin(thetaaz)*sin(thetac)*sin(thetael)*sin(

            thetax)*sin(thetaz) - cos(phic)*cos(thetaaz)*cos(thetax)*

            cos(thetaz)*sin(psic)*sin(thetay) + cos(phic)*cos(thetaaz)

            *sin(psic)*sin(thetax)*sin(thetaz) - cos(phic)*cos(thetax)

            *cos(thetay)*sin(psic)*sin(thetaaz) + cos(psic)*cos(

            thetaaz)*cos(thetac)*cos(thetael)*cos(thetax)*cos(thetay)
```

```
           - cos(psic)*cos(thetaaz)*cos(thetax)*cos(thetaz)*sin(phic
           )*sin(thetac)*sin(thetay) + cos(psic)*cos(thetaaz)*sin(
           phic)*sin(thetac)*sin(thetax)*sin(thetaz) - cos(psic)*cos(
           thetac)*cos(thetael)*cos(thetax)*cos(thetaz)*sin(thetaaz)*
           sin(thetay) + cos(psic)*cos(thetac)*cos(thetael)*sin(
           thetaaz)*sin(thetax)*sin(thetaz) + cos(psic)*cos(thetac)*
           cos(thetax)*sin(thetael)*sin(thetay)*sin(thetaz) + cos(
           psic)*cos(thetac)*cos(thetaz)*sin(thetael)*sin(thetax) -
           cos(psic)*cos(thetax)*cos(thetay)*sin(phic)*sin(thetaaz)*
           sin(thetac) + cos(thetaaz)*cos(thetax)*cos(thetay)*sin(
           phic)*sin(psic)*sin(thetael) - cos(thetael)*cos(thetax)*
           sin(phic)*sin(psic)*sin(thetay)*sin(thetaz) - cos(thetael)
           *cos(thetaz)*sin(phic)*sin(psic)*sin(thetax) - cos(thetax)
           *cos(thetaz)*sin(phic)*sin(psic)*sin(thetaaz)*sin(thetael)
           *sin(thetay) + sin(phic)*sin(psic)*sin(thetaaz)*sin(
           thetael)*sin(thetax)*sin(thetaz)

Ly    =    - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetay)*cos(thetaz
           ) + cos(phic)*cos(psic)*sin(thetaaz)*sin(thetay) - cos(
           phic)*cos(thetaaz)*sin(psic)*sin(thetac)*sin(thetael)*sin(
           thetay) - cos(phic)*cos(thetael)*cos(thetay)*sin(psic)*sin
           (thetac)*sin(thetaz) - cos(phic)*cos(thetay)*cos(thetaz)*
           sin(psic)*sin(thetaaz)*sin(thetac)*sin(thetael) - cos(psic
               )*cos(thetaaz)*sin(phic)*sin(thetael)*sin(thetay) - cos(
               psic)*cos(thetael)*cos(thetay)*sin(phic)*sin(thetaz) - cos
               (psic)*cos(thetay)*cos(thetaz)*sin(phic)*sin(thetaaz)*sin(
               thetael) + cos(thetaaz)*cos(thetac)*cos(thetael)*sin(psic)
               *sin(thetay) + cos(thetaaz)*cos(thetay)*cos(thetaz)*sin(
               phic)*sin(psic)*sin(thetac) + cos(thetac)*cos(thetael)*cos
               (thetay)*cos(thetaz)*sin(psic)*sin(thetaaz) - cos(thetac)*
               cos(thetay)*sin(psic)*sin(thetael)*sin(thetaz) - sin(phic)
               *sin(psic)*sin(thetaaz)*sin(thetac)*sin(thetay)
My     =       - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetax)*sin(thetaz
               ) - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetaz)*sin(
               thetax)*sin(thetay) - cos(phic)*cos(psic)*cos(thetay)*sin(
               thetaaz)*sin(thetax) + cos(phic)*cos(thetaaz)*cos(thetay)*
               sin(psic)*sin(thetac)*sin(thetael)*sin(thetax) + cos(phic)
               *cos(thetael)*cos(thetax)*cos(thetaz)*sin(psic)*sin(thetac
               ) - cos(phic)*cos(thetael)*sin(psic)*sin(thetac)*sin(
               thetax)*sin(thetay)*sin(thetaz) - cos(phic)*cos(thetax)*
               sin(psic)*sin(thetaaz)*sin(thetac)*sin(thetael)*sin(thetaz
               ) - cos(phic)*cos(thetaz)*sin(psic)*sin(thetaaz)*sin(
               thetac)*sin(thetael)*sin(thetax)*sin(thetay) + cos(psic)*
               cos(thetaaz)*cos(thetay)*sin(phic)*sin(thetael)*sin(thetax
               ) + cos(psic)*cos(thetael)*cos(thetax)*cos(thetaz)*sin(
              phic) - cos(psic)*cos(thetael)*sin(phic)*sin(thetax)*sin(
              thetay)*sin(thetaz) - cos(psic)*cos(thetax)*sin(phic)*sin(
              thetaaz)*sin(thetael)*sin(thetaz) - cos(psic)*cos(thetaz)*
              sin(phic)*sin(thetaaz)*sin(thetael)*sin(thetax)*sin(thetay
              ) - cos(thetaaz)*cos(thetac)*cos(thetael)*cos(thetay)*sin(
              psic)*sin(thetax) + cos(thetaaz)*cos(thetax)*sin(phic)*sin
              (psic)*sin(thetac)*sin(thetaz) + cos(thetaaz)*cos(thetaz)*
              sin(phic)*sin(psic)*sin(thetac)*sin(thetax)*sin(thetay) +
              cos(thetac)*cos(thetael)*cos(thetax)*sin(psic)*sin(thetaaz
              )*sin(thetaz) + cos(thetac)*cos(thetael)*cos(thetaz)*sin(
              psic)*sin(thetaaz)*sin(thetax)*sin(thetay) + cos(thetac)*
              cos(thetax)*cos(thetaz)*sin(psic)*sin(thetael) - cos(
              thetac)*sin(psic)*sin(thetael)*sin(thetax)*sin(thetay)*sin
              (thetaz) + cos(thetay)*sin(phic)*sin(psic)*sin(thetaaz)*
              sin(thetac)*sin(thetax)

Ny    =       - cos(phic)*cos(psic)*cos(thetaaz)*cos(thetax)*cos(thetaz
              )*sin(thetay) + cos(phic)*cos(psic)*cos(thetaaz)*sin(
              thetax)*sin(thetaz) - cos(phic)*cos(psic)*cos(thetax)*cos(
              thetay)*sin(thetaaz) + cos(phic)*cos(thetaaz)*cos(thetax)*
              cos(thetay)*sin(psic)*sin(thetac)*sin(thetael) - cos(phic)
              *cos(thetael)*cos(thetax)*sin(psic)*sin(thetac)*sin(thetay
              )*sin(thetaz) - cos(phic)*cos(thetael)*cos(thetaz)*sin(
psic)*sin(thetac)*sin(thetax) - cos(phic)*cos(thetax)*cos(
thetaz)*sin(psic)*sin(thetaaz)*sin(thetac)*sin(thetael)*
sin(thetay) + cos(phic)*sin(psic)*sin(thetaaz)*sin(thetac)
*sin(thetael)*sin(thetax)*sin(thetaz) + cos(psic)*cos(
thetaaz)*cos(thetax)*cos(thetay)*sin(phic)*sin(thetael) -
cos(psic)*cos(thetael)*cos(thetax)*sin(phic)*sin(thetay)*
sin(thetaz) - cos(psic)*cos(thetael)*cos(thetaz)*sin(phic)
*sin(thetax) - cos(psic)*cos(thetax)*cos(thetaz)*sin(phic)
*sin(thetaaz)*sin(thetael)*sin(thetay) + cos(psic)*sin(
phic)*sin(thetaaz)*sin(thetael)*sin(thetax)*sin(thetaz) -
cos(thetaaz)*cos(thetac)*cos(thetael)*cos(thetax)*cos(
thetay)*sin(psic) + cos(thetaaz)*cos(thetax)*cos(thetaz)*
sin(phic)*sin(psic)*sin(thetac)*sin(thetay) - cos(thetaaz)
*sin(phic)*sin(psic)*sin(thetac)*sin(thetax)*sin(thetaz)
 + cos(thetac)*cos(thetael)*cos(thetax)*cos(thetaz)*sin(
psic)*sin(thetaaz)*sin(thetay) - cos(thetac)*cos(thetael)*
sin(psic)*sin(thetaaz)*sin(thetax)*sin(thetaz) - cos(
thetac)*cos(thetax)*sin(psic)*sin(thetael)*sin(thetay)*sin
(thetaz) - cos(thetac)*cos(thetaz)*sin(psic)*sin(thetael)*
sin(thetax) + cos(thetax)*cos(thetay)*sin(phic)*sin(psic)*
sin(thetaaz)*sin(thetac)
```

```
Lz    =    cos(phic)*cos(thetaaz)*cos(thetac)*sin(thetael)*sin(thetay
           ) + cos(phic)*cos(thetac)*cos(thetael)*cos(thetay)*sin(
           thetaz) + cos(phic)*cos(thetac)*cos(thetay)*cos(thetaz)*
           sin(thetaaz)*sin(thetael) - cos(thetaaz)*cos(thetac)*cos(
           thetay)*cos(thetaz)*sin(phic) + cos(thetaaz)*cos(thetael)*
           sin(thetac)*sin(thetay) + cos(thetac)*sin(phic)*sin(
           thetaaz)*sin(thetay) + cos(thetael)*cos(thetay)*cos(thetaz
           )*sin(thetaaz)*sin(thetac) - cos(thetay)*sin(thetac)*sin(
           thetael)*sin(thetaz)

Mz    =    - cos(phic)*cos(thetaaz)*cos(thetac)*cos(thetay)*sin(
           thetael)*sin(thetax) - cos(phic)*cos(thetac)*cos(thetael)*
           cos(thetax)*cos(thetaz) + cos(phic)*cos(thetac)*cos(
           thetael)*sin(thetax)*sin(thetay)*sin(thetaz) + cos(phic)*
           cos(thetac)*cos(thetax)*sin(thetaaz)*sin(thetael)*sin(
           thetaz) + cos(phic)*cos(thetac)*cos(thetaz)*sin(thetaaz)*
           sin(thetael)*sin(thetax)*sin(thetay) - cos(thetaaz)*cos(
           thetac)*cos(thetax)*sin(phic)*sin(thetaz) - cos(thetaaz)*
           cos(thetac)*cos(thetaz)*sin(phic)*sin(thetax)*sin(thetay)
            - cos(thetaaz)*cos(thetael)*cos(thetay)*sin(thetac)*sin(
           thetax) - cos(thetac)*cos(thetay)*sin(phic)*sin(thetaaz)*
           sin(thetax) + cos(thetael)*cos(thetax)*sin(thetaaz)*sin(
           thetac)*sin(thetaz) + cos(thetael)*cos(thetaz)*sin(thetaaz
```

```
                    )*sin(thetac)*sin(thetax)*sin(thetay) + cos(thetax)*cos(
                    thetaz)*sin(thetac)*sin(thetael) - sin(thetac)*sin(thetael
                    )*sin(thetax)*sin(thetay)*sin(thetaz)

Nz     =            - cos(phic)*cos(thetaaz)*cos(thetac)*cos(thetax)*cos(
                    thetay)*sin(thetael) + cos(phic)*cos(thetac)*cos(thetael)*
                    cos(thetax)*sin(thetay)*sin(thetaz) + cos(phic)*cos(thetac
                    )*cos(thetael)*cos(thetaz)*sin(thetax) + cos(phic)*cos(
                    thetac)*cos(thetax)*cos(thetaz)*sin(thetaaz)*sin(thetael)*
                    sin(thetay) - cos(phic)*cos(thetac)*sin(thetaaz)*sin(
                    thetael)*sin(thetax)*sin(thetaz) - cos(thetaaz)*cos(thetac
                    )*cos(thetax)*cos(thetaz)*sin(phic)*sin(thetay) + cos(
                    thetaaz)*cos(thetac)*sin(phic)*sin(thetax)*sin(thetaz) -
                    cos(thetaaz)*cos(thetael)*cos(thetax)*cos(thetay)*sin(
                    thetac) - cos(thetac)*cos(thetax)*cos(thetay)*sin(phic)*
                    sin(thetaaz) + cos(thetael)*cos(thetax)*cos(thetaz)*sin(
                    thetaaz)*sin(thetac)*sin(thetay) - cos(thetael)*sin(
                    thetaaz)*sin(thetac)*sin(thetax)*sin(thetaz) - cos(thetax)
                    *sin(thetac)*sin(thetael)*sin(thetay)*sin(thetaz) - cos(
                    thetaz)*sin(thetac)*sin(thetael)*sin(thetax)
```

DISTRIBUTION LIST

Commander
Armament Research, Development and Engineering Center
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCAR-IMI-I (5)
SMCAR-FSF-RC (15)
Picatinny Arsenal, NJ 07806-5000

Commander
U.S. Army Armament, Munitions and Chemical Command
ATTN: AMSMC-GCL(D)
Picatinny Arsenal, NJ 07806-5000

Administrator
Defense Technical Information Center
ATTN: Accessions Division (12)
Cameron Station
Alexandria, VA 22304-6145

Director
U.S. Army Material Systems Analysis Activity
ATTN: SMCCR-MSI
Aberdeen Proving Ground, MD 21005-5066

Commander
Chemical Research, Development and Engineering Center
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCCR-MSI
Aberdeen Proving Ground, MD 21010-5423

Commander
Chemical Research, Development and Engineering Center
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCCR-RSP A
Aberdeen Proving Ground, MD 21010-5423

Director
Ballistic Research Laboratory
ATTN: AMXBR-OD-ST
Aberdeen Proving Ground, MD 21005-5066

Chief
Benet Weapons Laboratory, CCAC
Armament Research, Development and Engineering Center
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCAR-CCB-TL
Watervliet, NY 12189-5000

Commander
U.S. Army Armament, Munitions and Chemical Command
ATTN: SMCAR-ESP-L
Rock Island, IL 61299-6000

Director
U.S. Army TRADOC Systems Analysis Activity
ATTN: ATAA-SL
White Sands Missle Range, NM 88002